VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

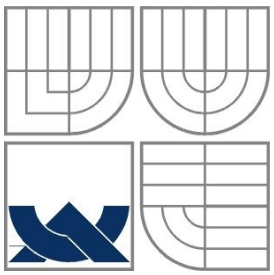
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ PODOBNÝCH FOTOGRAFIÍ

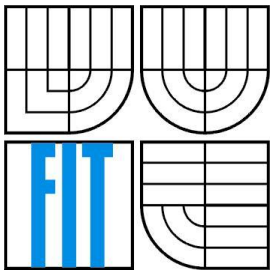
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. ŠTĚPÁN ROSA
AUTHOR

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHLEDÁVÁNÍ PODOBNÝCH FOTOGRAFIÍ

SIMILAR PHOTO SEARCHING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ŠTĚPÁN ROSA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VÍTĚZSLAV BERAN

BRNO 2010

Abstrakt

Tato práce popisuje cestu k realizaci aplikace, ve které si uživatel vybere databázi fotografií, se kterou bude pracovat a zadá systému fotografii. Ten mu pomocí vizuálního slovníku nalezne nejpodobnějších fotografie z této databáze a na základě statistické analýzy textových popisků těchto fotografií mu nabídne vhodnou formou popisky pro dotazovanou fotografii.

Abstract

This paper describes the way to realization such an application, where a user chooses a photo database to working with and enters a photo into the system. The system using a visual vocabulary finds the most similar photos from the database and offers tags of the searched photo with a suitable form based on the tag statistical analysis of this photo.

Klíčová slova

zpracování obrazu, lokální detektory, MSER, SIFT, normalizace afinní oblasti, vizuální slovník, vyhledávání, relevance textových popisků, automatický návrh popisků, mrak štítků

Keywords

preprocessing, local deskriptor, MSER, SIFT, affine region normalization, visual vocabulary, matching, tag relevance, automatic photo tagging, tag cloud

Vyhledávání podobných fotografií

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……

Štěpán Rosa

24. května 2010

Poděkování

Děkuji panu Ing. Vítězslavu Beranovi za metodické vedení, ochotnou spolupráci a cenné rady.

Obsah

1 Úvod

V dnešní moderní době se počítače staly nedílnou součástí našeho života. Pomáhají nám, usnadňují nám život. Dalo by se říct, že na nich závisí fungování novodobé společnosti. Umí to a chovají se podle toho, co je naučíme.

Jednou z věcí, kterou ještě stále nejsme schopni počítače dokonale naučit, je vidět tak, jak vidíme my. Člověk se rozhlédne kolem sebe a bez problému dokáže rozpoznávat objekty, tvary, lidi, vidět mezi vjemy souvislosti, odhadovat vzdálenosti, poznávat bez problému místa z různých pohledů a to vše s neuvěřitelnou rychlostí a lehkostí. Pro počítač je však obraz pouze pole čísel získaných ze senzorů. Když si představíme, že bychom místo barev a tvarů kolem sebe viděli obrovskou matici čísel, uvědomíme si, jak složitou úlohou počítačové vidění je.

Cílem této diplomové práce je prostudovat dostupnou literaturu týkající se metod získávání příznaků z obrazu a jejich statistické analýzy, dále pak seznámit se s metodami statistické analýzy textových informací a na základě nastudovaných informací navrhnout systém, který získá z fotografií příznaky a provede jejich statistickou analýzu společně s textovými informacemi u jednotlivých obrázků. V dalším kroku pak pořídit sadu obrazových dat s textovými popisky tak, aby mohla být implementována navržená experimentální aplikace, která zobrazí výsledek a umožní jeho korekci. V neposlední řadě provést experimenty zkoumající výkonnost a stabilitu navrženého systému, diskutovat nedostatky a navrhnout případná vylepšení, vytvořit plakát reprezentující řešení.

Jedná se o téma, které je zajímavé a užitečné. Vyhledáním vizuálně podobných fotografií, které jsou oanotované textovou informací nám dává možnost ulehčit uživateli práci v tom smyslu, že natrénovaný systém mu podle zvolených nastavení informaci nabídne vhodné popisky.

Následující kapitola čtenáře provede problematikou, jak nalézt a popsat části obrazu tak, aby bylo možné je opakovaně nalézat nezávisle na místě pohledu na scénu i za různých světelných podmínek, což je jeden z předpokladů pro vyhledávání podobných fotografií.

Třetí kapitola ho seznámí s tím, jak těchto popisů využít pro reprezentaci vizuálního obsahu fotografie tak, aby bylo možné efektivně provádět vyhledávání pomocí metriky vzájemné podobnosti. Bude zde vysvětlen princip vizuálního slovníku a jeho tvorby.

Ve 4. kapitole se nachází část věnovaná textovým popiskům, jejich analýze a problémům, které s ní souvisí. Dále se také popisuje mrak štítku, což je jeden z vhodných způsobů, jak popisky zobrazit.

Pátá kapitola obsahuje návrh požadovaného systému. Systém bude tvořen dvěma částmi. V první části bude navrženo online vyhledávání, v druhé bude popsáno trénování systému.

Kroky potřebné k realizaci navrženého řešení jsou obsaženy v šesté kapitole. Je zde uvedeno, jakým způsobem byla získána oanotovaná data a jak bylo třeba tato data upravit. Vyskytuje se zde popis vytvoření vizuálního slovníku, korpusu dat, a demonstrační aplikace.

Experimenty na datové sadě čítající 1000 fotografií v 10 třídách jsou obsahem sedmé kapitoly. Zde bude sledováno, jak dobře systém fungoval při různě zadaných parametrech při vytváření jeho částí. Bude provedeno vyhodnocení, jaké procento z nejbližších sousedů bylo ze stejné třídy jako dotazovaná fotografie. Třídy fotografií jsou tvořeny tak, aby klíčové slovo použité při jejich vyhledání bylo relevantní, tj. aby daný objekt na fotografii byl přítomen, aby byl celý vidět, aby fotografie nebyla rozmazaná a byla to fotografie z reálného světa, ne malba či kresba.

Závěrečná kapitola obsahuje v krátkosti shrnutí dosažených výsledků a návrhy na možné vylepšení a další rozšíření projektu.

Tato diplomová práce navazuje na semestrální projekt stejného názvu řešený v předchozím semestru. Ze semestrálního projektu je využit s úpravami tento úvod, dále pak 2. – 5. kapitola, přičemž kapitoly 3, 4 a 5 byly rozšířeny. Dále je využita sada obrazových dat s textovými popisky.

2 Příznaky

Abychom mohli pracovat s obrazy je třeba je nějakým způsobem popsat. K tomuto účelu složí příznaky. Příznaky mohou být globální – spočítané nad celým obrazem, nebo lokální – získané z části obrazu. V této práci bude pozornost věnována výhradně lokálním příznakům[1]. Na začátku kapitoly se zmíním o nejznámějších detektorech, které v obrazu naleznou zajímavé oblasti, které se používají pro popis obrazu. Poté bude následovat bližší popis detektoru maximálně stabilních extrémních oblastí - MSER a deskriptoru SIFT pro jejich dobré vlastnosti. Poslední část je věnována normalizaci detekované oblasti.

2.1 Současný stav

Rozhodujícím a obtížným krokem k plně automatické rekonstrukci 3D scén je nalezení spolehlivých korespondencí mezi dvěma obrazy jedné scény získaných z libovolného úhlu pohledu s možností použití různých kamer a za rozdílných světelných podmínek. Klíčovou otázkou je zde volba elementů, jejichž korespondence je hledaná. Pro složitější úkoly si nevystačíme s použitím translace a rotace k aproximaci lokálních deformací obrazu a je potřeba vytvořit plně afinní model. Korespondence nemohou být ustaveny obyčejným porovnáním oblastí, protože tvar oblastí se mění afinní transformací. Ve většině obrazů existují oblasti, které mohou být detekovány s vysokou opakovatelností, díky svým charakteristickým, neměnným a stabilním vlastnostem. A právě tyto oblasti nazývané charakteristické oblasti mohou sloužit jako elementy, na nichž se zkoumají korespondence [1].

Detekce oblastí kovariantních s třídou transformací má velkou oblast využití. Využití naleznou např. při vyhledávání obrázků ve velkých databázích, vyhledávání ve videu, rozpoznávání založených na modelech, tvoření panorámat, lokalizaci robotů a dalších. Požadavkem na takové oblasti je, aby pro různé pohledy na stejnou scénu obsahovali shodné části. Tedy, aby se tvar těchto oblastí automaticky přizpůsoboval a byl projekcí stejné části 3D povrchu. Pan Mikolajczyk a spol. ([2]) provedli testování 6 detektorů na datech, která obsahovala strukturované scény i scény s texturami, na něž byly aplikovány různé transformace: změna pohledu, změna měřítka, změna osvětlení, rozmazání a JPG komprese a došel k závěrům, že u všech detektorů dochází pozvolna k poklesu výkonu se vzrůstající změnou úhlu pohledu a že neexistuje detektor, který by byl nejlepší jak u různých typů scén, tak při různých transformacích.

V mnoha případech dosahoval nejlepších výsledků MSER, následovaný Herrisem-afinním. Zjistil také, že MSER a IBR fungují dobře na obrázcích obsahujících homogenní oblasti se zřetelnými hranicemi. Hessian-afinní a Harris-afinní ve srovnání s ostatními detektory poskytují více oblastí než

[1] Pro úplnost uvádím, že globální příznaky mohou být např. histogram gradientů, barev, diskrétní kosinová transformace (DCT) a další.

ostatní detektory, což je užitečné při porovnávání scén obsahujících stejné objekty, které jsou z části zakryté, ale tak když obsahují rušivé elementy. Detektor založený na hranách je vhodný pro scény obsahující průsečíky hran. Detektor význačných oblastí pracoval dobře při rozpoznávání objektů. Protože detektory se doplňují, může být k dosažení nejlepších výsledků použito jejich kombinace [2]. To s sebou nutně nese zvýšení doby výpočtu, je tedy vhodné přizpůsobit jejich výběr požadavkům aplikace. Následuje krátký popis zmiňovaných detektorů.

Harris-Afinní

Tento detektor je založený na detekci bodů v odpovídajících měřítkách pomocí Laplaciánu. Okolí bodů ohraničené elipsou ale i samotný bod jsou určeny pomocí matice druhých momentů gradientu intenzity. Tato matice je také často nazývána auto-korelační a popisuje distribuci gradientu v okolí bodu:

$$M = \mu(\boldsymbol{x}, \sigma_I, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(\boldsymbol{x}, \sigma_D) & I_x I_y(\boldsymbol{x}, \sigma_D) \\ I_x I_y(\boldsymbol{x}, \sigma_D) & I_y^2(\boldsymbol{x}, \sigma_D) \end{bmatrix} \quad (1)$$

Lokální změny obrazu jsou vypočítány pomocí Gaussova konvolučního jádra s odchylkou σ_D, které představuje diferenciační měřítko. Tyto změny jsou poté ještě průměrovány s okolím bodu konvolucí s Gaussovým oknem s integrační odchylkou σ_I (integrační měřítko). Hledají se takové body, pro které jsou obě vlastní čísla této matice velká, poněvadž tyto body přestavují rohové body, které jsou stabilní v libovolných světelných podmínkách a jsou reprezentanty obrazu.

Ke zjištění charakteristického měřítka se používá Laplacův operátor. Charakteristické měřítko je zvoleno na základě extrému odezev na různá měřítka. Velikost oblasti je tak vybrána nezávisle na rozlišení obrazu. Když jsou určeny počátečný body a jejich charakteristické měřítko, tak se iterativním způsobem na základě matice druhých momentů odhaduje tvar afinní oblasti. Podle odhadu je provedena normalizace na kruhovou oblast. Takto se pokračuje, dokud vlastní čísla matice druhých momentu nejsou shodná [2].

Výhodou tohoto detektoru je, že není citlivý na 2D posuny a rotace, malé změny osvětlení a úhly pohledu. Je výpočetně nenáročný. Na druhou stranu není invariantní vůči větším změnám měřítka, úhlu pohledu a významným změnám kontrastu [3]. Pro obrázek s rozlišením 800x640 detekuje přibližně1000 oblastí [4].

Hessian-Afinní

Pracuje na podobném principu jako Herris-Afinní. Má velkou odezvu na skvrny a vyvýšeniny v intenzitě. Je založen na Hessově matici:

$$H = H(\boldsymbol{x}, \sigma_D) = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} I_{xx}(\boldsymbol{x}, \sigma_D) & I_{xy}(\boldsymbol{x}, \sigma_D) \\ I_{xy}(\boldsymbol{x}, \sigma_D) & I_{yy}(\boldsymbol{x}, \sigma_D) \end{bmatrix} \quad (2)$$

Druhé derivace detekují oblasti podobné těm, které se získají Laplacovým operátorem. Funkce založená na determinantu Hessovi matice však penalizuje velmi dlouhé struktury, pro které je druhá derivace v určitém směru velmi malá. Lokální maxima determinantu indikují přítomnost skvrn. Odhad charakteristického měřítka a tvar afinní oblasti se určuje stejně jako v případě detektoru Harris-Afinní [2].

Vyhodnocením bylo zjištěno, že Hessian-Afinní pracuje jako druhý nejlepší po MSERu. Má dobrou odezvu na scény obsahující textury, ale dosahuje dobrých výsledků i u některých strukturovaných scén jako například u budov. Stejně jako Herris-Afinní detekuje hodně spíše menších oblastí. Typicky určuje více spolehlivých oblastí než Harris-Afinní [5].

Detektor založený na hranách – EBR

Detektor využívá hrany v obraze. Na začátku se nalezne Harrisův rohový bod **p** a sousední hrana získaná Cannyho hranovým detektorem. Ke zvýšení robustnosti vůči změnám měřítka jsou tyto základní příznaky extrahovány ve více měřítkách. Princip je ilustrován na Obrázek 1. Od rohu se v obou směrech podél hrany vzdalují body p_1 a p_2. Relativní rychlost jejich pohybu je určena rovností relativních afinně invariantních parametrů l_1 a l_2:

$$l_i = \int abs\left(\left|\boldsymbol{p_i^{(1)}}(s_i)\,\boldsymbol{p} - \boldsymbol{p_i}(s_i)\right|\right) ds_i \tag{3}$$

kde s_i je libovolný parametr křivky (v obou směrech), $\boldsymbol{p_i^{(1)}}(s_i)$ první derivace v bodě $\boldsymbol{p_i}(s_i)$ podle s_i, $abs()$ absolutní hodnota a $|...|$ determinant. Tato podmínka předepisuje, že oblast ohraničená spojnicí bodů **p**, $\mathbf{p_1}$ a hranou zůstane shodná s oblastí mezi spojnicí bodů **p**, $\mathbf{p_2}$ a hranou. Když bude odkazováno na $l_1 = l_2$, může se jednoduše použít l.

Pro každou hodnotu l body $\boldsymbol{p_1}(l)$ a $\boldsymbol{p_2}(l)$ společně s rohem **p** definují rovnoběžník $\Omega(l)$: rovnoběžník roztáhnutý vektory $\boldsymbol{p_1}(l) - \boldsymbol{p}$ a $\boldsymbol{p_2}(l) - \boldsymbol{p}$. To přináší jednorozměrnou řadu oblastí rovnoběžníkového tvaru jako funkci l. Z této řady vybereme jeden (nebo málo) rovnoběžníků, pro které následující fotometrické veličiny textury prochází extrémem.

$$\mathrm{Inv}_1 = abs\left(\frac{|\mathbf{p_1} - \mathbf{p_g} \quad \mathbf{p_2} - \mathbf{p_g}|}{|\mathbf{p} - \mathbf{p_1} \quad \mathbf{p} - \mathbf{p_2}|}\right) \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)2}} \tag{4}$$

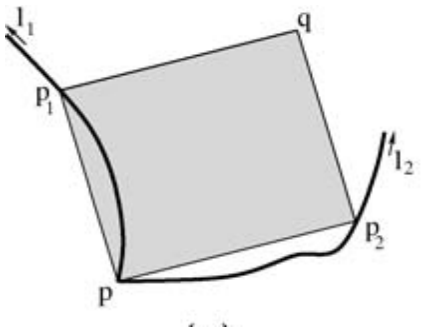
$$\mathrm{Inv}_2 = abs\left(\frac{|\mathbf{p} - \mathbf{p_g} \quad \mathbf{q} - \mathbf{p_g}|}{|\mathbf{p} - \mathbf{p_1} \quad \mathbf{p} - \mathbf{p_2}|}\right) \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)2}} \tag{5}$$

kde

$$M_{pq}^{n} = \int_{\Omega} I^{n}(x,y) x^{p} y^{q} dydy \tag{6}$$

$$\mathbf{p_g} = \left(\frac{M_{10}^{1}}{M_{00}^{1}}, \frac{M_{01}^{1}}{M_{00}^{1}} \right) \tag{7}$$

kde M_{pq}^{n} je n-tý řád, $(p+q)$-tý stupeň momentu nad oblastí $\Omega(l)$, $\mathbf{p_g}$ těžiště oblasti váhované intenzitou $I(x,y)$ a $\mathbf{q}$ protější roh rohu $\mathbf{p}$ rovnoběžníku [2].



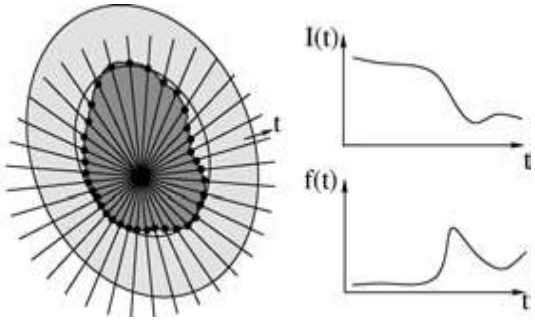
Obrázek 1: Detektor založený na hranách [2]

Detektor založený na extrémech intenzity - IBR

Tento detektor vychází, jak je již z názvu patrné, z extrémů intenzit, které jsou detekovány ve více měřítkách. Podél paprsků vedených z těchto bodů je pro každý paprsek zkoumána funkce intenzity:

$$f_I(t) = \frac{abs(I(t) - I_0)}{\max\left(\frac{\int_0^t abs(I(t) - I_0) dt}{t}, d \right)} \tag{8}$$

kde t je libovolný parametr podél paprsku, $I(t)$ intenzita v bodě t, I_0 hodnota intenzity v extrému a d mále číslo přidané z důvodu zabránění dělení nulou (viz Obrázek 2 dole). Bod pro daný paprsek, kde funkce dosahuje extrému je invariantní vůči afinním geometrickým a lineárním fotometrickým transformacím. Maxima se typicky dosahuje v místě, kde intenzita náhle roste nebo klesá. Funkce $f_I(t)$ je již sama o sobě invariantní, avšak kvůli robustnímu výběru se používají body, kde funkce dosahuje extrémů. Všechny body, ve kterých funkce $f_I(t)$ podél paprsků vycházejících ze stejného lokálního extrému dosahuje maxima, jsou spojeny a uzavírají tak afinně kovariantní oblast. Ta je většinou tvořena nepravidelným tvarem, a proto je nahrazena elipsou, která má stejné tvarové momenty až do druhého řádu [2].



Obrázek 2: Detektor založený na extrémech intenzit [2]

Detektor význačných oblastí

Detektor je založený na distribuční funkci pravděpodobnosti (pdf) hodnot intenzity vypočítané přes oblast ohraničenou elipsou. Detekce probíhá ve dvou krocích. Nejprve se nad třemi parametry elipsy centrované na daném pixelu vyhodnotí entropie pdf. Tento výpočet se provede pro každý pixel. Množina extrémů entropie přes různá měřítka je spolu s odpovídajícími parametry elipsy zaznamenána. Tato množina tvoří kandidáty význačných oblastí. Dalším krokem je seřazení všech kandidátů podle velikosti derivace pdf s ohledem na měřítko. Ponecháno je P oblastí s nejvyšší hodnotou derivace. Oblast ohraničená elipsou $\mathcal{E}$ se středem v pixelu **x** je parametrizována jejím měřítkem s (které určuje hlavní osu), její orientací θ (hlavní osy), a poměrem mezi hlavní a vedlejší osou λ. Pdf intenzity je vypočítaná nad $\mathcal{E}$. Entropie $\mathcal{H}$ je dána:

$$\mathcal{H} = -\sum_I p(I) \log p(i) \tag{9}$$

Množina extrémů nad měřítky v $\mathcal{H}$ je vypočítána pro parametry s, θ, λ pro každý pixel obrazu. Pro každý extrém je derivace pdf $p(I; s, \theta, \lambda)$ v měřítku s vypočítána

$$\mathcal{W} = \frac{s^2}{2s-1} \sum_I \left| \frac{\partial p(I; s, \theta, \lambda)}{\partial s} \right| \tag{10}$$

a význačnost $\mathcal{Y} = \mathcal{H}\mathcal{W}$. Oblasti jsou seřazeny podle význačnosti $\mathcal{Y}$ [2].
Výkonnost tohoto detektoru je ve srovnání s ostatními nižší především, protože detekuje menší počet bodů. Avšak při úlohách rozpoznávání 3D objektů může být výkonný [6].

2.2 Detektor maximálně stabilních extrémních oblastí - MSER

Jedná se o detektor, který detekuje skvrny. Navrhnul jej Matas a spol. k nalezení korespondencí částí obrazu dvou obrazů s různým bodem pohledu. Důvodem bližšího popisu toho detektoru a jsou výsledky, kterých dosáhnul v testu detektorů. Výstup detektoru je vidět na Obrázek 3. Nejlépe se vypořádal se změnou pohledu jak u strukturovaných scén, tak u scén s texturami. U změn měřítka skončil na druhém místě. Nejhůře však dopadl při rozmazání obrazu, jelikož je více citlivý na tento typ transformace. Poněvadž při vyhlazených hranicích oblastí je proces jejich nalezení méně přesný. Při změnách osvětlení prokázal největší opakovatelnost [7].

Extrémní oblasti mají dvě požadované vlastnosti. Množina je uzavřená vůči spojitým (a tedy projektivním) transformacím obrazových souřadnic. To znamená, že je afinně invariantní a nezáleží na tom, jestli je obraz zdeformovaný nebo zkosený. Je také uzavřená vůči monotónním transformacím intenzit obrazu. Tedy fotometrické změny při detekci nehrají roli, je tedy jedno jestli se jedná o vnitřní nebo vnější osvětlení, jestli je slunečný den nebo oblačný či noční doba [7].

Jak je uvedeno v [1], maximálně stabilní extrémní oblasti jsou definovány výhradně extrémní vlastností funkce intenzity v dané oblasti a na její vnější hranici. Neformálně by mohl být koncept vysvětlen následujícím způsobem. Představme si všechny možné prahy šedotónového obrázku *I*. Pixelům, které mají hodnotu menší než je práh, přiřadíme černou barvu a těm které mají hodnotu vyšší nebo rovnou prahu bílou barvu. Kdybychom zobrazili film takto prahovaných obrázků I_t, se snímkem *t* odpovídajícím prahu *t*, viděli bychom nejprve bílý obrázek. Následně by se objevily a zvětšovaly černé skvrny odpovídající lokálním minimům intenzity. Postupně by docházelo ke slučování skvrn. Poslední obrázek sekvence by byl celý černý. Množina všech spojených komponent všech snímků filmu tvoří množinu všech maximálních oblastí. Minimální oblasti mohou být získány inverzí intenzity *I* a spuštěním stejného procesu. Následuje formální definice.

Obraz I je zobrazení $I: \mathcal{D} \subset \mathbb{Z}^2 \rightarrow S$. Extrémní oblasti v obrazu jsou dobře ohraničené, když:

1. S je úplně uspořádána, tj. existuje reflexivní, antisymetrická a tranzitivní binární relace $\leq$.
2. Je definována relace sousednosti $A \subset \mathcal{D} \times \mathcal{D}$.

Oblast Q je spojitá podmnožina $\mathcal{D}$, tj. pro každé $p, q \in Q$existuje sekvence $p, a_1, a_2, \dots, a_n, q$ a $pAa_1, a_iAa_{i+1}, a_nAq$.

(Vnější) hranice oblasti $\partial Q = \{q \in \mathcal{D} \setminus Q : \exists p \in Q : qAp\}$, tj. hranice ∂Q oblasti Q je množina pixelů sousedících nejméně s jedním pixelem z Q, ale nepatřící Q.

Extrémní oblast $Q \subset \mathcal{D}$ je taková oblast, kde pro všechny $p \in Q, q \in \partial Q : I(p) > I(q)$ (oblast s maximální intenzitou) nebo $I(p) < I(q)$ (oblast s minimální intenzitou).

Maximálně stabilní extrémní oblast (MSER). Nechť $\mathcal{Q}_1, \dots, \mathcal{Q}_{i-1}, \mathcal{Q}_i, \dots$ je sekvence vnořených extrémních oblastí, tj. $\mathcal{Q}_i \subset \mathcal{Q}_{i+1}$. Extrémní oblasti $\mathcal{Q}_{i^*}$ je maximálně stabilní když a jen když $q(i) = |\mathcal{Q}_{i+\Delta} \setminus \mathcal{Q}_{i-\Delta}|/|\mathcal{Q}_i|$ má lokální minimum v i^* (|.| označuje kardinalitu). $\Delta \in S$ je parametr metody.

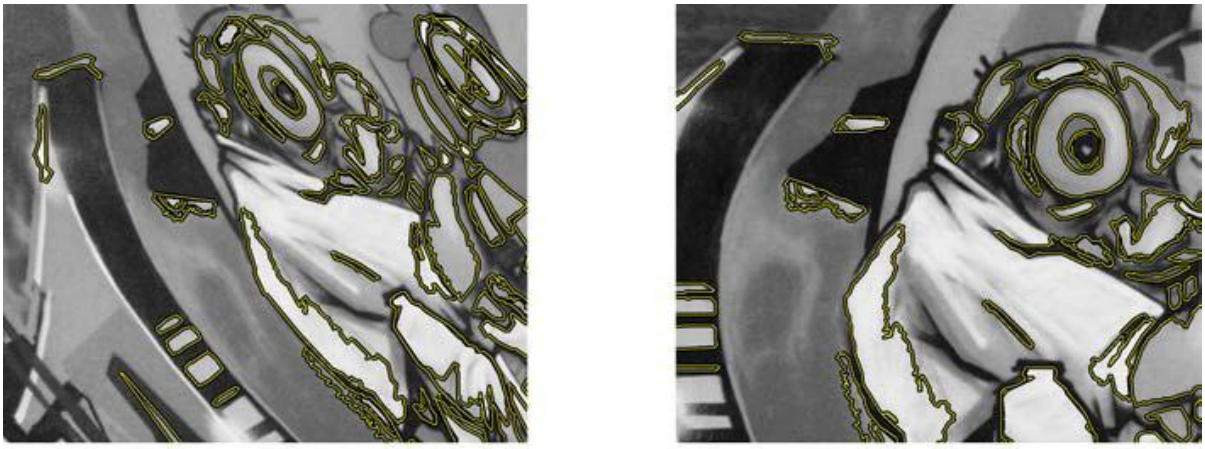
V mnoha obrazech je v určitých oblastech lokální binarizace stabilní přes velký rozsah prahů. Takové oblasti jsou středem zájmu, poněvadž představují vlastnosti jako:

- Invarianci vůči afinním transformacím intenzit obrazu.
- Kovarianci zachovávající sousednost (spojitě).
- Stabilitu, protože jsou vybrány jen extrémní oblasti, které jsou téměř stejné přes rozsah prahů.
- Detekci v mnohonásobném měřítku, aniž by se použilo vyhlazování, detekovány jsou velmi drobné i velmi velké struktury
- Výpočet množiny všech extrémních regionů je $O(n \log\log n)$, kde n označuje počet pixelů v obraze.

Výpočet extrémních oblastí probíhá následujícím způsobem. Nejprve jsou pixely seřazeny podle intenzity. Výpočetní složitost tohoto kroku je $O(n)$, za podmínky, že rozsah hodnot S obrazu je malý, tj. typicky $\{0, \dots, 255\}$, jelikož třídění může být implementováno algoritmem Binsort. Po třídění jsou pixely umísťovány do obrazu (buď v klesajícím, nebo rostoucím pořadí) a pomocí efektivního algoritmu union-find je udržován seznam spojených komponent a oblastí, které zabírají. V praxi je algoritmus velmi rychlý.

Zpracování produkuje datové struktury ukládající oblast každé spojené komponenty jako funkci intenzity. Na spojení dvou komponent je pohlíženo jako na ukončení existence menší komponenty a vložení všech pixelů, které obsahovala do větší z nich. Nakonec se jako práh produkující maximálně stabilní extrémní oblast vezme úroveň intenzity, při které relativní změna oblasti jako funkce relativní změny prahu dosahuje lokálního minima. Na výstupu je každý MSER reprezentován pozicí lokálního minima (nebo maxima) intenzity a prahem.

Každá extrémní oblast je spojená komponenta prahovaného obrazu. Nicméně se nehledá globální nebo optimální práh, nýbrž se testují všechny prahy a vyhodnocuje se stabilita spojené komponenty. Výstupem MSER detektoru není binární obraz. Pro některé části obrazů existují mnohonásobné stabilní prahy. V tom případě je výstupem systém vnořených podmnožin.



Obrázek 3: Ukázka oblastí detekovaných detektorem MSER [2]

2.3 SIFT

Tento algoritmus pro detekci a popis lokálních příznaků publikoval v roce 1999 David Lowe [8]. Získané příznaky mají mnoho vlastností, které jsou vhodné pro porovnání rozdílných obrazů objektů či scén. Příznaky jsou invariantní vůči změnám měřítka, rotacím a částečně vůči změně osvětlení a změně 3D pohledu. Jsou dobře lokalizované v prostorové i frekvenční doméně. Umožňují identifikovat objekty, i když jsou v přeplněné scéně či částečně zakryté. Z typických obrázků je možné získat efektivním algoritmem velký počet příznaků, které hustě pokrývají obraz plným rozsahem měřítek a lokalizací. Příznaky jsou vysoce rozlišující, což umožňuje, aby byl příznak s vysokou pravděpodobností správně spárován i ve velkých databázích. Používá se kaskádový přístup, ve kterém jsou více výpočetně náročné operace aplikovány jen na místa, která splnily počáteční test. Typické hlavní kroky výpočtu pro generování množiny příznaků jsou následující:

1. **Hledání extrému v prostoru měřítek** – První fáze výpočtu hledá přes všechny měřítka a pozice v obrazu pomocí diferencí Gaussových křivek (diference-of-Gaussian) potenciálně zajímavé body, které jsou invariantní vůči měřítku a orientaci.
2. **Pozice klíčového bodu** – U těchto bodů jsou pomocí detailnějšího provedení určeny pozice a měřítko. Klíčové body jsou vybrány na základě jejich stability.
3. **Určení orientace** – Každému klíčovému bodu je přiřazena jedna nebo více orientací na základě směru gradientů v lokální části obrazu. Všechny budoucí operace pro každý příznak jsou vykonány na obrazových datech, která byla transformována relativně vůči přiřazené orientaci, měřítku a pozici. Tím je zajištěna invariance vůči těmto transformacím.
4. **Deskriptor klíčového bodu** – Lokální obrazové gradienty v oblasti kolem klíčového bodu ve zvoleném měřítku jsou transformovány do reprezentace, která připouští výraznou úroveň deformace lokálního tvaru i změnu osvětlení.

Dále bude detailně popsána pouze fáze určení orientace a získání deskriptoru, poněvadž tyto fáze budou využity při realizaci aplikace.

Aby bylo dosaženo invariance vůči natočení, tak je každému klíčovému bodu na základě jeho vlastností přiřazena odpovídající orientace a deskriptor je vyjádřen relativně vůči této orientaci. Na základě měřítka klíčového bodu je v pyramidě obrazů vybrán Gaussovou funkcí vyhlazený obraz L, který je nejblíže zvolenému měřítku. Pro každý pixel $L(x, y)$ v tomto měřítku je vypočítána velikost a směr gradientu pomocí následujících rovnic:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{11}$$

$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{12}$$

Ze směru gradientů pixelů uvnitř oblasti kolem klíčového bodu se vytvoří histogram orientací. Tento histogram má 36 binů a pokrývá rozsah 360 stupňů orientací. Každý vzorek, který se přidá do histogramu, je váhovaný velikostí jeho gradientu a Gaussovým kruhovým oknem se směrodatnou odchylkou σ, která se rovná 1,5násobku měřítka klíčového bodu.

Vrcholy v histogramu orientací odpovídají dominantním směrům lokálních gradientů. Určí se nejvyšší vrchol v histogramu a na základě toho je klíčovému bodu přiřazena tato orientace. Pro stejný účel se také použijí všechny lokální vrcholy histogramu, které dosahují 80% výšky nejvyššího. Pro každý takový vrchol se kvůli zlepšení přesnosti interpoluje jeho pozice pomocí paraboly proložené 3 hodnotami, které jsou v histogramu vrcholu nejblíže. Takto může vzniknout vícenásobný klíčový bod se stejnou pozicí, se stejným měřítkem ale odlišnou orientací. Vícenásobná orientace je přiřazena asi jen 15% klíčových bodů, avšak tyto značně přispívají ke zlepšení stability porovnávání.

Na základě předchozích operací jsou již každému klíčovému bodu přiřazeny kromě pozice a měřítka také orientace. Tyto parametry zavádí opakovatelný lokální 2D systém souřadnic pro popis lokální oblasti obrazu, a proto poskytují invarianci vůči těmto parametrům. Dalším krokem je výpočet deskriptoru pro tuto oblast, který je vysoce rozlišující přesto však co nejvíce invariantní vůči změnám v osvětlení nebo v 3D pohledu.

Výpočet samotného deskriptoru je demonstrován na Obrázek 4. Podle měřítka bodu je vybrán obraz s odpovídající rozostřením. Za účelem dosažení invariance orientace jsou souřadnice deskriptoru i orientace gradientů otočeny relativně vůči orientaci klíčového bodu. Velikosti gradientů oblasti jsou v každém bodě váhovány pomocí Gaussova okna se σ rovnou polovině šířky okna deskriptoru. Na obrázku je Gaussovo okno zobrazeno pomocí kruhu. Důvodem použití Gaussova okna je vyhnutí se náhlým změnám deskriptoru zapříčiněným malou změnou pozice okna a také

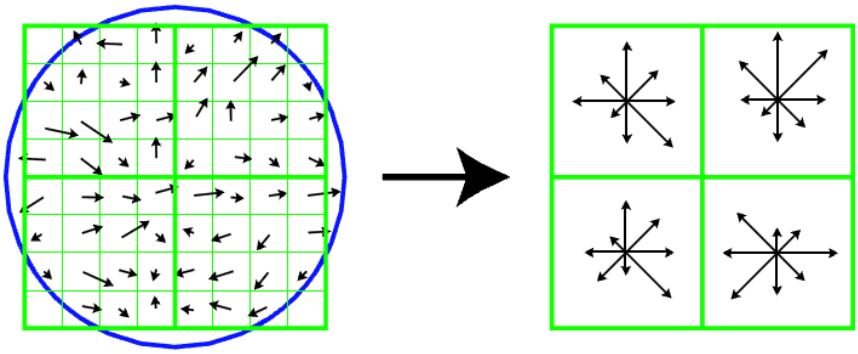
zmenšení váhy gradientů, které jsou daleko od středu deskriptoru, poněvadž ty jsou nejvíce ovlivněny chybnou registrací.

Obrázek 4 vpravo zobrazuje deskriptor. Skládá se ze 4 histogramů (po 8 binech), které shrnují obsahy podoblastí o rozměru 4x4. Každý histogram tvoří 8 šipek, jejichž délka odpovídá součtu váhovaných velikostí gradientů blízko tohoto směru. Vzorek s gradientem vlevo může být posunut až o 4 pozice, přičemž bude stále přispívat do stejného histogramu.

Aby se eliminovaly hraniční efekty, ve kterých se náhle změní deskriptor vlivem geometrické deformace, kdy vzorek přejde hladce z jednoho histogramu do druhého nebo z jedné orientace do druhé, používá se k distribuci hodnoty každého vzorku gradientu trilineární interpolace. Každý příspěvek do binu je násoben s váhou $1 - d$, kde d je vzdálenost vzorku od centrální hodnoty binu. Vzdálenost mezi sousedními biny se rovná jedné.

Deskriptor je vytvořený z vektoru obsahujícího hodnoty všech záznamů histogramů korespondujících s délkou šipek. Na obrázku je zobrazeno pole histogramů orientací o rozměru 2x2, avšak pan Lowe experimenty ukázal, že nejlepší výsledky jsou dosaženy pro pole 4x4, kde každý histogram obsahuje 8 binů. Tedy vektor příznaků tvoří 4x4x8 = 128 hodnot pro každý klíčový bod.

Nakonec je ještě vektor příznaků upraven tak, aby byly redukovány účinky změny osvětlení. Nejprve je vektor příznaků normalizován na jednotkovou délku. Změna kontrastu obrazu, při které je každá hodnota pixelu násobena konstantou, vynásobí gradienty stejnou konstantou, takže při normalizaci vektoru je tato změna zrušena. Změna jasu, při které je ke každé hodnotě pixelu přidána konstanta neovlivní hodnoty gradientu, protože tyto hodnoty jsou spočteny z rozdílů pixelů. Proto je deskriptor invariantní vůči afinním změnám v osvětlení. Nicméně kvůli saturaci kamery nebo změnám v osvětlení, které ovlivňují 3D povrchy s různými orientacemi různým množstvím, mohou nastat nelineární změny v osvětlení. Tyto účinky mohou pro některé gradienty zapříčinit velké změny velikosti, avšak změny orientace jsou méně pravděpodobné. Aby se redukoval vliv velkých velikostí gradientů, jsou hodnoty jednotkového vektoru prahovány tak, aby nebyly větší než hodnota 0,2 a poté je znovu provedena normalizace vektoru. Důsledkem toho je, že nebude tak důležitá shoda velikosti velkých gradientů a že se zvýší důraz na distribuci orientací. Hodnotu 0,2 určil pan Lowe experimentálně na obrázcích obsahujících stejné 3D objekty různě osvětlené [9].



Obrázek 4: Deskriptor SIFT [9]

2.4 Normalizace

Detektor MSER detekuje oblasti libovolného tvaru a velikosti. Abychom z této oblasti získali afinně kovariantní oblast ohraničenou elipsou, kterou bude možné normalizovat vůči geometrickým deformacím a získat tak kruhovou oblast [2], jejíž ohraničující čtverec dáme na vstup SIFT deskriptoru, je třeba provést následující kroky.

Oblast detekovanou MSERem použijeme pro výpočet kovarianční matice (matice centrálních momentů druhého řádu) [10]. Nejprve se určí podle rovnice 13 těžiště oblasti $\mathbf{m} = [\bar{x}, \bar{y}]$.

$$\bar{x} = \frac{M_{10}}{M_{00}}, \bar{y} = \frac{M_{01}}{M_{00}} \tag{13}$$

$$M_{ij} = \sum_x \sum_y x^i y^j \, I(x,y) \tag{14}$$

kde M_{ij} je obrazový moment řádu $(i+j)$, $I(x,y)$ intenzita pixelu šedotónového obrazu, x, y souřadnice pixelu.

Centrální momenty se vypočítají:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p \, (y-\bar{y})^q I(x,y) \tag{15}$$

$$\mu'_{kl} = \frac{\mu_{kl}}{\mu_{00}} \tag{16}$$

A konečně kovarianční matice

$$\mathrm{cov}[I(\bar{x},\bar{y})] = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix} = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \tag{17}$$

kde $\bar{x}, \bar{y}, a, b, c$ definují afinní oblast $a(x-\bar{x})^2 + 2b(x-\bar{x})(y-\bar{y}) + c(y-\bar{y})^2 = 1$, souřadnice [0,0] se předpokládá v levém horním rohu [11, 12].

Pro normalizaci eliptického tvaru použijeme singulární rozklad [13] kovarianční matice $\mathbf{C} = \mathbf{RDR}^T$ (s determinantem **R** větším než nula), kde **D** je v případě kovarianční matice vektor vlastních čísel a **R** matice vlastních vektorů. Transformační matice je

$$\mathbf{x} = s\mathbf{A}\hat{\mathbf{x}} + \mathbf{m}, \text{ pro } \mathbf{A} = 2\mathbf{R}\mathbf{D}^{\frac{1}{2}} \tag{18}$$

V normalizované oblasti by měl být bod na pozici $\hat{\mathbf{x}}$, získán z původního obrazu na pozici $\mathbf{x}$. Parametr s představuje změnu měřítka, udává, kolikrát bude oblast, nad níž se bude provádět převod do normalizovaného tvaru, větší než oblast popsaná parametry $\bar{x}, \bar{y}, a, b, c$. Před vzorkováním se oblast rozmaže Gaussovým konvolučním jádrem se σ_i. σ_i je určena výrazem $\sigma_i = \frac{hs}{N_s}$, kde h je menší osa aproximující elipsy, N_s počet vzorků v normalizované části obrazu. Vzorkování na pozici $\hat{\mathbf{x}} \in [-1,1]^2$ je provedeno pomocí bilineární interpolace, následované dalším rozmazáním pomocí Gaussova konvolučního jádra se σ_p [10].

3 Vizuální slovník

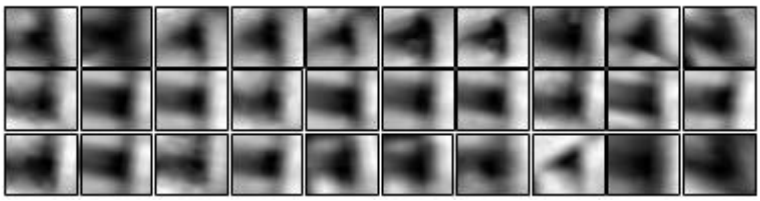
Tato kapitola se zabývá otázkou, jak vhodně reprezentovat vizuální obsah fotografie, tak aby bylo možné nalézt fotografii s podobným obsahem z velkého množství kandidátů efektivním způsobem. Bude uveden koncept vizuálního slovníku, postup jeho vytvoření, vzdálenostní metriky použité pro vyhledávání, vyhledávácí struktury a na závěr popis obrazu pomocí vizuálního slovníku.

3.1 Úvod

Myšlenka vizuálního slovníku pochází z článku [14], kde zkoumali, zda můžou být použity techniky z přirozeného zpracování jazyka a z oblasti získávání informací k popisu snímků ve videu tak, aby bylo možné ve filmu vyhledávat části obrazu, které nás zajímají, takovým způsobem, jakým jsme zvyklí při vyhledávání informací na internetu, tedy rychle a přesně. Například nás můžou zajímat snímky v obraze, kde se vyskytuje logo nějaké firmy, či určitý objekt. Potom není problém vypočítat např. jaké procento z délky filmy je logo vidět, atd. Principu vizuálního slovníku bylo také využito při on-line synchronizaci videa [15].

Princip vizuálního slovníku spočívá v tom, že lokální příznaky se nahradí identifikátorem vizuálních slov. Provede se výpočet histogramu výskytu slov ve fotografii (bag-of-words). Na tento histogram se aplikuje váhování. Vzniká tak vizuální popisek obrázku, který umožňuje rychlé vyhledávání podobných fotografií a poskytuje dobré výsledky.

Vizuální slovník je seznam středů shluků a jim přirazených popisků (identifikační číslo) vytvořených shlukovou analýzou provedenou nad prostorem deskriptorů příznaků. Na Obrázek 5 jsou vidět části obrazu, jejichž deskriptory odpovídají jednomu vizuálnímu slovu [14].



Obrázek 5: Části obrazu jež odpovídají jednomu vizuálnímu slovu [14]

3.2 Vytvoření slovníku

Prvním krokem při vytváření vizuálního slovníku je extrakce příznaků z trénovací sady obrázků. Všechny tyto příznaky se zanesou do prostoru příznaků, který se pomocí shlukové analýzy rozdělí na jednotlivé shluky. Shluková analýza je založená na algoritmu k-means, který je popsán níže.

K-means

1. Určí se počet shluků K.
2. Určí se počáteční body, které budou reprezentovat středy shluků $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_\mathrm{K}$. Výběr bodů je proveden buď náhodně, nebo se vezme prvních K bodů nebo se použije nějaká heuristika.
3. Každý bod $\mathbf{x_i}$ je přiřazen shluku, k jehož středu má nejmenší vzdálenost $y_i = \arg\min \|\mathbf{x_i} - \boldsymbol{\mu}\mathrm{k}2, k \in 1,\ldots,K$
4. Provede se přepočítání středů $\boldsymbol{\mu}_\mathrm{k} = \frac{1}{\mathrm{N}_{y_i=\mathrm{k}}} \sum_{i:y_i=k} \mathbf{x}_i$
5. Opakují se kroky 3, 4 dokud se mění pozice středů shluků [16].

3.3 Vyhledávací struktury

Při hledání nejbližšího středu shluku se používá několik variant v závislosti na velikosti výsledného slovníku. Tyto varianty jsou zobrazeny na Obrázek 6.

Naivní k-means

Používá se, když je velikost slovníku malá (do 1000 slov). Časová složitost je $O(kN)$, kde k je velikost slovníku a N je počet trénovacích vektorů příznaků [15].

Hierarchický k-means

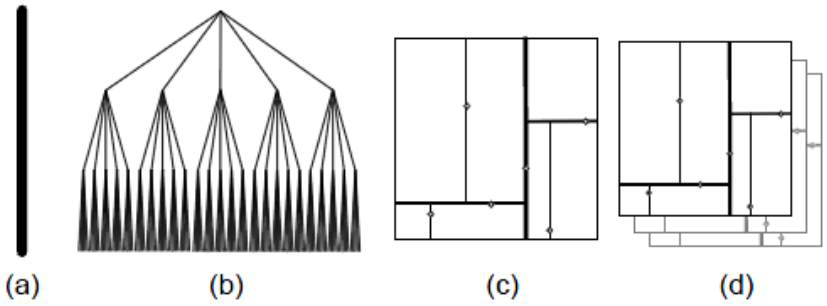
Představuje jeden ze způsobů, jak redukovat časovou složitost naivního k-means. Vyhledávání se realizuje průchodem stromovou strukturou. Tímto způsobem je docíleno časové složitosti $O(N.\log \mathrm{k})$ [15].

Kd-strom

Další přístup pro redukci časové složitosti, kdy je nahrazeno hledání nejbližšího souseda kd-stromem. Kd-strom je technika dělení prostoru, kdy řezné roviny jsou vždy kolmé na některou ze souřadnicových os, přičemž orientace řezů se pravidelně střídají [17].

Aproximovaný k-means

Tato metoda místo hledání nejbližších sousedů mezi body v prostoru deskriptoru příznaků a středy shluků, používá metodu aproximovaného nejbližšího souseda a les 8 náhodných kd-stromů vytvořených nad středy shluků na začátku každé iterace. Časová složitost je $O(N.\log \mathrm{k})$ [18].

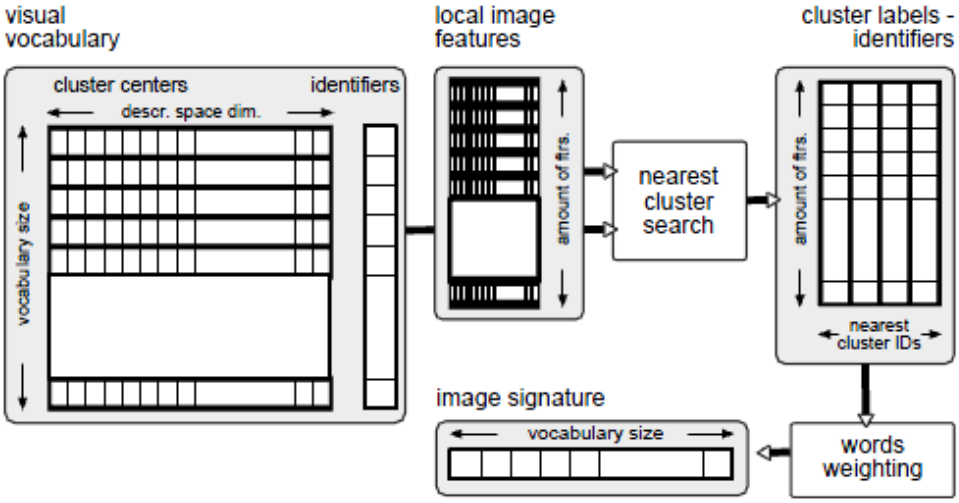


Obrázek 6: Vyhledávací strategie – (a) naivní sekvenční přístup, (b) hierarchický, (c) kd-strom, (d) náhodný les [15]

3.4 Vizuální popisek obrázku (VpO)

Vizuální popisek obrázku je kolekce váhovaných vizuálních slov reprezentujících obsah obrazu. Můžeme sena něj dívat také jako na vektor frekvencí výskytu vizuálních slov v obraze. Pokud váhy vyjadřují pouze přítomnost vizuálního slova, hovoříme o set-of-words, v opačném případě se jedná o bag-of-words [15].

Proces vytvoření vizuálního popisku obrázku je zobrazen na Obrázek 7. Ve fotografii jsou nejprve detekovány a následně popsány lokální příznaky. Pomocí vizuálního slovníku je pro každý deskriptor příznaků nalezeno k-nejbližších vizuálních slov (nejbližších středů shluků). Na základě různých váhovacích schémat je pro každé vizuální slovo vypočítána váha s jakou přispívá do VpO na pozici identifikátoru daného vizuálního slova [15].



Obrázek 7: Proces získání vizuálního popisku obrázku [15]

3.5 Váhování slov

Místo pouhé četnosti výskytu vizuálních slov, se pro vytvoření VpO často používá váhovaní jednotlivých složek.

Tf-idf

Jedno z váhování, které se používá také při textovém vyhledávání je známé jako frekvence slova – inverzní frekvence dokumentů (term frequency – inverse dokument frequency). Kde frekvence slova odráží entropii slova vůči každému dokumentu (fotografii), na rozdíl od inverzní frekvence dokumentů, která snižuje váhu slov, které se objevují v dokumentech příliš často.

$$tf - idf(w) = \frac{|d(w)|}{|d|} \cdot \log\left(\frac{|D|}{|D(w)|}\right) \tag{19}$$

kde d je dokument (množina slov), $|d|$ je počet slov v d a $d(w)$ je počet výskytů slova w v dokumentu d, D je datová množina všech dokumentů a $D(w)$ je množina dokumentů obsahujících slovo w [15].

Soft-assignment I

Metoda jemného přiřazení (soft-assignment) přiřadí jednomu deskriptoru několik nejbližších vizuálních slov v prostoru příznaků, aby zohlednila vzdálenost mezi vizuálními slovy a deskriptorem. V seřazeném seznamu k nejbližších vizuálních slov se pro vyjádření vzdálenosti používá exponenciální funkce

$$w = \exp\left(-\frac{d^2}{2\sigma^2}\right) \tag{20}$$

kde d je vzdálenost deskriptoru od středu shluku, w výsledná váha a σ hodnota, která se v praxi volí tak, aby podstatná váha byla přiřazena jen malému počtu slov (pro $k = 3; \sigma = 6250$) [15].

Soft-assignment II

Přístup je založen na pořadí přiřazeného slova

$$w = \frac{1}{2^{i-1}} \tag{21}$$

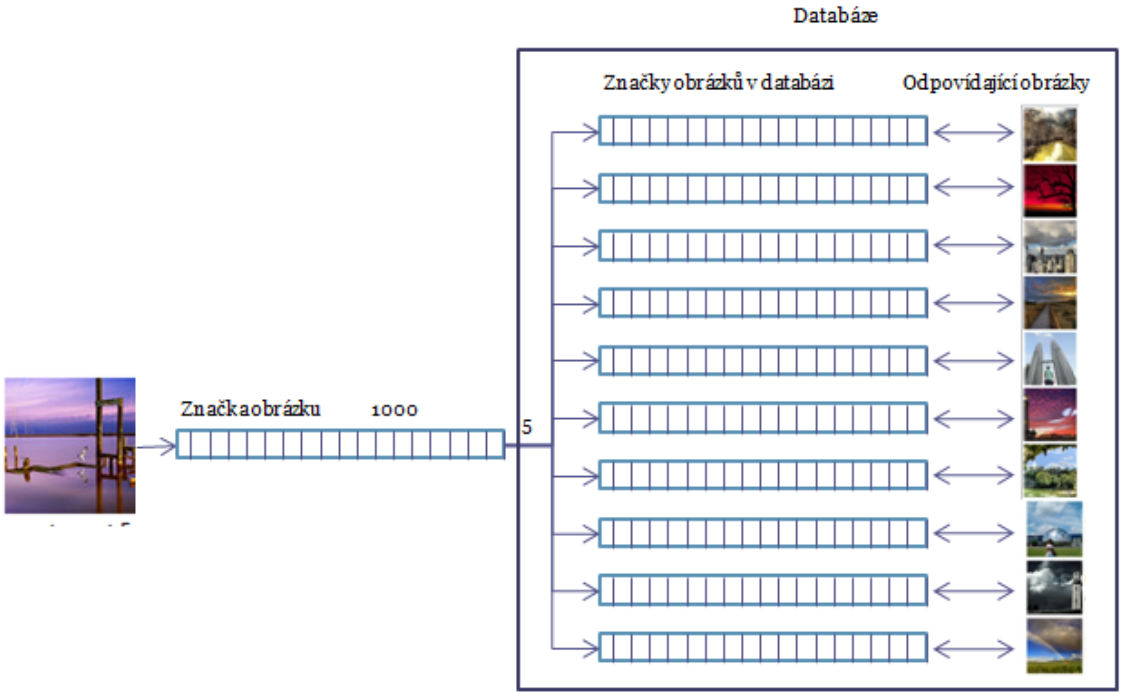
kde i je i-tý nejbližší soused [15], empiricky autoři určili, že je rozumné hledat 4 nejbližší sousedy [19].

3.6 Míra podobnosti

Pro dotazovanou fotografii se vypočítá VpO. U všech fotografií v databázi byl VpO dopředu spočítán. Databáze tak tvoří korpus dat. Situace je znázorněna na Obrázek 8: Znázorněný vyhledání pomocí kosinové vzdálenost VpO dotazu a VpO obrázků v korpusu. Jedna z nejpoužívanějších technik je kosinová vzdálenost mezi dotazovaným vektorem **q** a vektory všech dokumentů **d** v databázi

$$sim(\mathbf{q}, \mathbf{d}) = \frac{\mathbf{q} \cdot \mathbf{d}}{|\mathbf{q}||\mathbf{d}|} \tag{22}$$

kde · je skalární součin a |... | je velikost vektoru [15].



Obrázek 8: Znázorněný vyhledání pomocí kosinové vzdálenost VpO dotazu a VpO obrázků v korpusu

4 Textové popisky

Předpokladem pro dobře fungující automatizovaný textový popis vizuálního obsahu fotografie je spolehlivé vyhledání podobných fotografií. Pokud je tento předpoklad splněn, další krok představuje získání textových popisků z nejbližších fotografií a jejich analýza, při které hraje důležitou roli relevance jednotlivých popisků. Ovšem v reálné situaci můžeme také narazit na problém s tzv. sémantickou mezerou, kdy vizuálně podobný obsah nemusí mít podobný sémantický význam. Následující podkapitoly čtenáře seznámí s používanými přístupy pro automatickou anotaci textových popisků, s problémem popisků, které se nevztahují k vizuální informaci, způsobem jak se s tímto vypořádat pomocí algoritmu pro určení relevance popisků a v závěru je uvedena zmínka o mraku štítků.

4.1 Automatická anotace

Existují metody, které jsou založené na modelech. Zde se natrénují klasifikátory, které jsou pak použité pro predikci relevantních popisků. Tyto metody jsou úspěšné na malých obrazových databázích. Avšak potenciálně neomezený počet slov, které uživatelé mohou zadat, obrovské rozmanitosti vizuální informace a nedostatek trénovacích dat mohou způsobit, že vytvořený model bude nespolehlivý a velmi všeobecný.

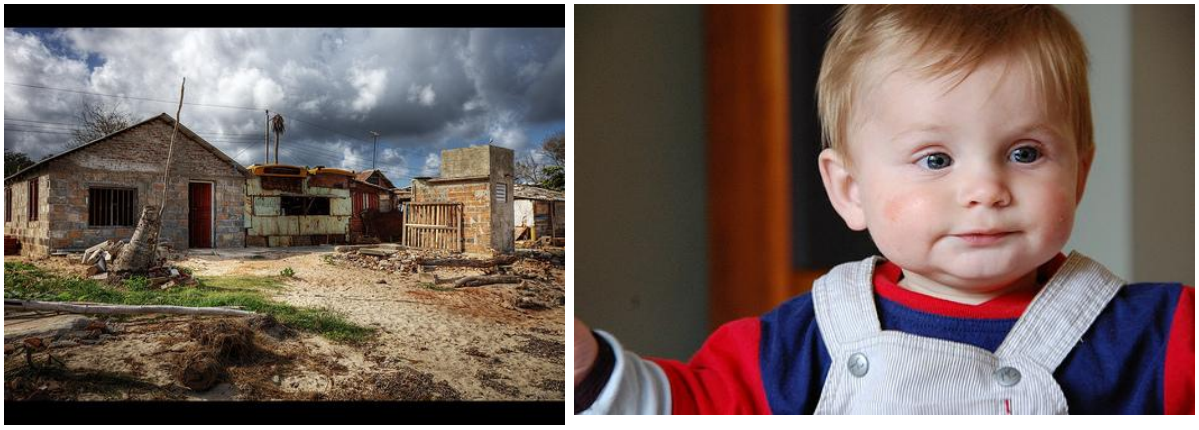
Jedním z možných přístupů k řešení tohoto problému jsou metody bez modelu, kde je cílem nalézt vizuálně podobné obrázky a použít jejich tagy k anotaci zadaného obrázku. Nicméně, zde se můžeme setkat se skutečností, že ne vždy má vizuálně podobný obsah také sémantickou podobnost. Tím tak mohou být do systému započítány nevýznamné tagy [20].

Tento problém může být vyřešen na základě poznatků, že ačkoliv samotný textový popis obrázků a jeho vizuální obsah mohou být sami o sobě matoucí, když se zkombinují dohromady, tak tomu tak není. Anotační problém můžeme rozdělit do dvou kroků. V prvním kroku vyhledat podobné fotografie a určit jeden správný textový popisek (např. eiffel). V druhém kroku pak vyhledat podobné fotografie pomocí vizuálního obsahu fotografie a určeného popisku, což s sebou přináší výhodu, že možná nejednoznačnost obrazových dat je redukována a vzniká tak větší pravděpodobnost, že budou nalezeny fotografie podobné, co se týče vizuálního i sémantického obsahu [21].

Dalším přístup může být určit relevanci jednotlivých textových popisků u všech fotografií v databázi. Tento přístup bude detailně popsán v následujících podkapitolách.

4.2 Relevance textových popisků

Problém, na který je třeba myslet při automatické anotaci fotografií je, že uživatelé, kteří díky tomu, že nejsou nějak svazováni, jak zadat popisek k obrázku zadávají různé štítky, které, jak je známo, jsou občas matoucí, nedostatečně vypovídající (ne-li vůbec) a dost personifikované. Příkladem je fotografie na Obrázek 9 vlevo, která byla vrácena jako jedna z nerelevantnějších odpovědí na klíčové slovo car v databázi flickr, ačkoliv na ní žádné auto není. Fotografie na Obrázek 9 vpravo je rovněž oštítkovaná jako car. Bude tedy nutné nějakým způsobem určit relevanci jednotlivých popisků.



Obrázek 9: Fotografie s popiskem car

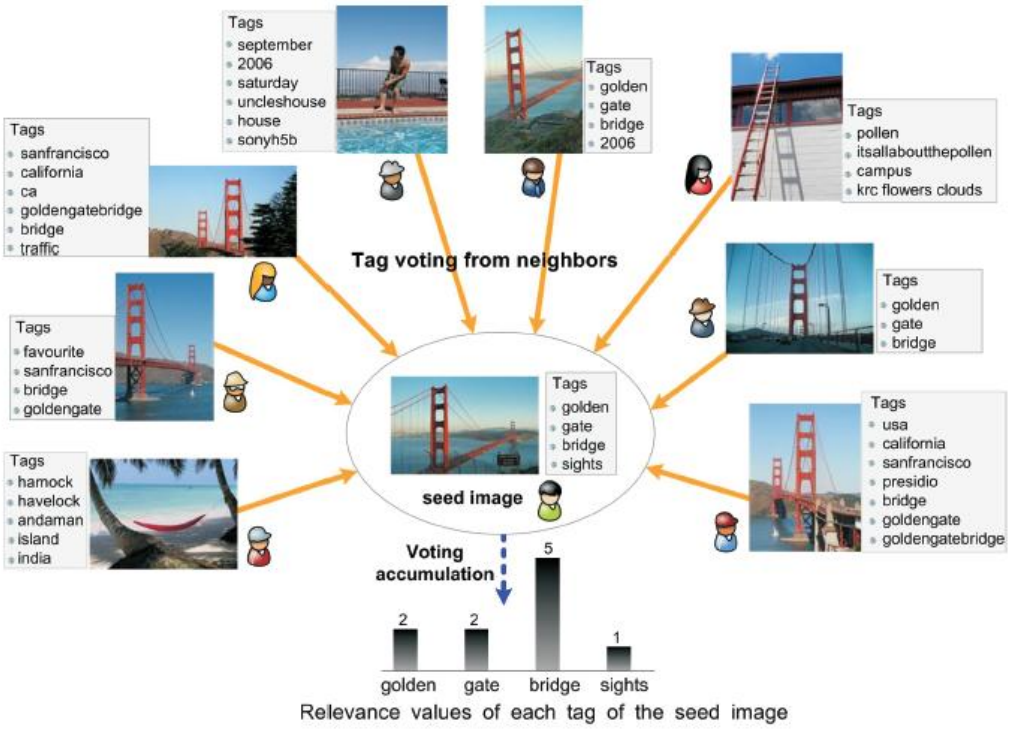
Míra relevance daného tagu vůči vizuálnímu obsahu je však otázkou subjektivního názoru. Je třeba najít nějaké objektivní kritérium pro určení relevance tagu, tak aby tento popisek „vyhovoval“ většině lidí. V článku [20] definovali, že tag je relevantní vůči obrázku, v případě, že objektivně popisuje aspekty vizuálního obsahu, tedy že obsah může být pomocí běžných znalostí jednoduše a konzistentně rozpoznám. Situaci stěžující fakt je, že jednotlivé tagy jsou většinou použity jednou na celý obrázek. Z toho vyplývá, že relevantní a nerelevatní popisek v obrázku nerozlišíme.

4.3 Výpočet relevance textových popisků

Algoritmus pro výpočet relevance u textových popisků je založen na myšlence, že jestliže jsou štítky dané fotografie relevantní, pak jistě, když si vyhledáme vizuálně podobné fotografie, budou tyto fotografie rovněž dané štítky obsahovat.

Pro danou fotografii I s textovými popisky, se nejprve nalezne k-nejpodobnějších fotografií. Poté jsou použity popisky z každé nalezené fotografie pro hlasování pro štítky u dotazované fotografie. Princip je zobrazen na Obrázek 10.

V případě, že různí lidé používají stejné popisky pro vizuálně podobné fotografie, budou pravděpodobně tyto popisky relevantní vizuálnímu obsahu, jež popisují. Proto je v algoritmu zakomponované omezení, že jestliže stejný uživatel oštítkoval fotografii, u níž se počítá relevance jednotlivých tagů a fotografii, která je v k-nejbližších, je tato fotografie vyloučena ze zvyšování relevance u dotazu.



Obrázek 10: Získání relevance tagu pomocí sousedů. Vizuálně podobní sousedé vůči zdrojové fotografii zvyšují relevanci jejich tagů. Např. 5 podobných fotografií jsou oštítkovány jako bridge, proto hodnota relevance tohoto tagu je 5. [20]

Algoritumus získání relevance v pseudokódu

```
Vstup: Fotografie I oštítkovaná uživatelem u.
Výstup: relevance každého tagu fotografie I

Najdi k nejbližších vizuálních sousedů fotografie I.
for each tag in tags
     relevance[tag] = 0;
for each fotografie in k-nejbližších
     if(jiný_uživatel_zadal_tagy_k_fotografii)
          for each tag in tag_obsažen_v_dotazu_i_v_fotografii
          relevance[tag]++;
```

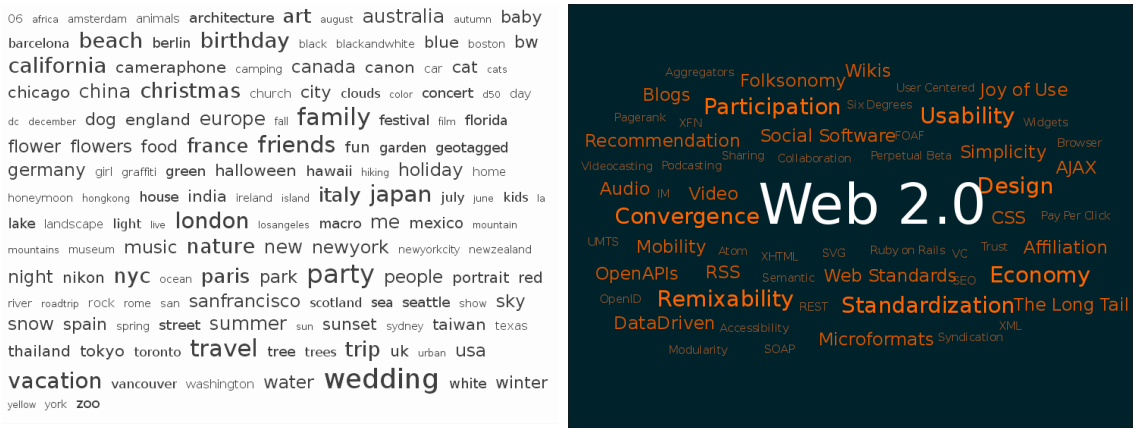
4.4 Stop list

Nad některými pro naše účely nesmyslnými textovými informacemi (např. popisek top50, aplusphoto atd.), který může bez problému obstát při relevanci textových popisků, je vhodným doplňkem použití stop listu, který tagy uvedené v tomto seznamu při analýze textových popisků nebude brát v potaz.

4.5 Mrak štítků

Pro zobrazení nejpodobnějších štítků se použije této techniky. Jak uvádí [22], mrak štítků nebo mrak slov je např. vizuální zobrazení uživatelsky generovaných popisků typicky používaných k popisu obsahu webových stránek. Popisky jsou obvykle jednoslovné názvy a většinou jsou seřazené abecedně. Význam popisku je ukázán velikostí písma nebo barvou. Uživateli je tak umožněno nalézt štítky abecedně i podle jejich oblíbenosti. První použití mraku štítků bylo na webovských stránkách Flickru [23], což jsou stránky pro sdílení fotografií. Příklad je vidět na Obrázek 11.

Nad touto technikou bylo provedeno několik studií použitelnosti. Výsledky jedné z nich jsou následující. Větší velikost písma štítků přitahuje větší pozornost, uživatelé spíše štítky přeletí pohledem, než aby je četli, štítky uprostřed mraku lákají více než ty u krajů, levý horní kvadrant získává větší pozornost než ostatní a konečně mraky štítků poskytují sub-optimální podporu při hledaní specifického štítku.



Obrázek 11: Mrak štítků

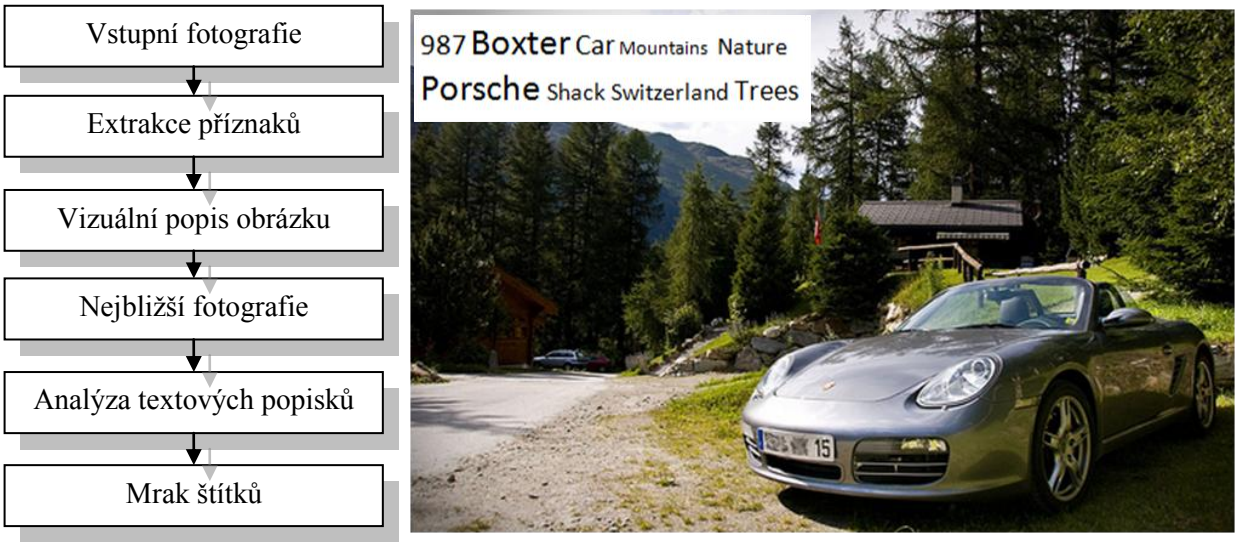
5 Návrh

Představa funkčnosti aplikace je následující. Uživatel si vybere konfigurační soubor, ve kterém budou všechny důležité parametry (jako např. korpus dat – tvořící VpO jednotlivých fotografií, soubor s textovými popisky, stop list, kolik nejbližších fotografií vzít v úvahu při navržení popisků, zda bude v úvahu brána relevance popisků, textový slovník) a fotografii, ke které bude chtít najít relevantní popisky. Systém nalezne vizuálně nejpodobnější fotografie a nabídne mu vhodné popisky. V první podkapitole je uveden návrh části pro online vyhledávání podobných fotografií a zobrazení výsledku, druhá podkapitola se zaměřuje na získání příznaků, vytvoření vizuálního slovníku a korpusu dat. Ve třetí kapitole je prezentován návrh předzpracování textových popisků.

5.1 Online vyhledávání

Blokové schéma postupu je vidět na Obrázek 12**Chyba! Nenalezen zdroj odkazů.** Do systému e načtena fotografie, které bychom chtěli automaticky přiřadit textové popisky. Nejprve se z obrazu získají příznaky.

V dalším kroku se provede výpočet VpO. Na základě VpO je prohledán korpus fotografií a pomocí kosinové vzdálenosti jsou určeny podobnosti. U prvních k-nejbližších kandidátů se provede analýza textových popisků. Vhodné popisky jsou uživateli zobrazeny v mraku štítků.



Obrázek 12 : Blokové schéma automatického návrhu popisku a navržený výstup programu

5.2 Fotografie

Trénovací data (fotografie) budou rozdělena do několika tříd. Název třídy bude reprezentovat popisek, který budou mít všechny fotografie nastavený a bude pro jejich obsah relevatní. Tedy fotografie jedné třídy budou mít podobný vizuální obsah a budou obsahovat zvolený typ objektu nebo scény. V úvahu budou brány nerozmazané fotografie, na kterých je zvolený objekt nebo scéna dobře viditelná. Bude se jednat o realné fotografie, nikoliv kreslené nebo malované obrázky. Objekty budou zobrazené z vnějšku nikoliv zevnitř [20]. Testovací fotografie se získají vyjmutím 10% fotografií z jednotlivých tříd a sadou několika náhodně stažených.

5.3 Tvorba vizuálního slovníku

Trénování systému spočívá ve vytvoření vizuálního slovníku. Ze všech trénovacích fotografií se vezmou příznaky a zanesou se do prostoru příznaků. Zvolí se velikost slovníku, která bude představovat počet vizuálních slov. S touto velikostí se bude experimentovat pro dosažený co nejlepších výsledků.

Vytvoření vizuálního slovníku bude probíhat podle teorie uvedené v podkapitole 3.2. Databáze pak bude převedena na reprezentaci pomocí VpO. Takto před-počítaná data umožní velmi rychlé vyhledávání.

5.4 Textová informace

Bude implementováno několik schémat pro váhování textových popisků. Nejjednodušší varianta bude pouze akumulovat výskyt popisků u vizuálně podobných fotografií. Lepší varianta bude využívat před-počítanou relevanci jednotlivých popisků.

Relevance se vypočítá podle teorie uvedené v kapitole 4. Soubor s textovými popisky bude obsahovat řádky v následujícím formátu.

```
Jméno_souboru          sum=x;tag1:y,tag2:z
```

Příklad:

```
/photos/1358453614.jpg sum=3,sun:1,sunset:1,sanjuanislands:0,water:0
```

Relevance bude ovlivnitelná parametrem představujícím počet nejbližších nalezených sousedů. Úměrně s tímto parametrem roste pravděpodobnost větší relevance daných popisků. Je důležité, aby

vyhledávání podobných fotografií fungovalo, co nejlépe, aby byla co nejpřesněji určena daná relevance.

Pro odstranění nesmyslných popisků bude použit stop list, popisky uvedené v tomto seznamu nebudou brány v úvahu při automatizovaném návrhu popisků.

Velikosti písma štítků v této práci budou vypočteny podle rovnice:

$$s_i = \frac{w_i - w_{min}}{w_{max} - w_{min}} (s_{max} - s_{min}) + s_{min} \quad (23)$$

kde s_i je velikost písma i-tého štítku, s_{min} je minimální velikost písma, s_{max} je maximální velikost písma, w_i je váha i-tého štítku, w_{max} je největší váha z množiny štítků a w_{min} je nejmenší váha z množiny štítků.

6 Realizace

Tato kapitola popisuje realizaci navrženého řešení. První podkapitola se zabývá získáním a úpravou dat potřebných pro natrénování a posléze vyhodnocení úspěšnosti systému. Druhá podkapitola popisuje knihovnu VPL, pomocí které byl systém natrénovaný. Třetí podkapitola se zabývá předzpracování textových popisků. Závěrečná podkapitola přibližuje implementaci experimentální aplikace.

6.1 Data

Ručně oanotovat dostatečnou sadu fotografií by časově příliš náročné. Místo toho bylo pomocí programu downloadr [24] (tento program umožňuje zapsat popisky, které byly v databazi, do souboru tvořícího fotografii) na jednom z nejznámějších úložišť a sice Flickru [22] vyhledáno a staženo přes 1000 fotografií.

Jednu skupinu z nich tvořili ručně vybraní vhodní kandidáti na klíčová slova *eiffel* (u této skupiny byl předpoklad nejlepšího vyhledání podobných fotografií, poněvadž se na všech fotografiích v této skupině vyskytuje eiffelova věž, tento předpoklad se také při experimentech potvrdil), *airplane*, *bicycle*, *bridge*, *building*, *butterfly*, *car*, *dog*, *flower* a *mountain*.

Protože velká část fotografií byla ve zbytečně vysokém rozlišení (2560x3915), které nepřináší pro náš účel žádné vylepšení, ba naopak, byly fotografie pomocí programu IrfanView [25] upraveny tak, aby delší strana měla maximálně 800 pixelů.

Poté pomocí programu ExifTool[26] byly do textového souboru extrahovány informace o názvu souboru fotografie a přišlusných popiscích ve formátu

```
Cesta  název_souboru_fotografie   tag, tag, tag, tag, tag.
```

6.2 Knihovna VPL

Po domluvě s vedoucím diplomové práce byla pro realizaci využita knihovna Video Processing Toolkit, jejímž je autorem. Knihovna poskytuje datové struktury, funkce a nástroje pro zpracování obrazu a videa, ukládání výsledků, extrakci příznaků a jejich analýzu, atd. [27] Využívá OpenCV [28] a databází PostgreSQL, nicméně umožňuje i práci bez databáze tím způsobem, že vytváří potřebné binární soubory na lokálním disku.

Následuje krátký popis použitých nástrojů:

Dataset

Tento nástroj umožňuje vytvoření datové struktury představující proces, ve kterém budou uloženy potřebné informace pro lokalizaci fotografií. Předpokládá na vstupu textový soubor ve formátu

```
Číslo sekvence (třídy); relativní_cesta_k_souboru_s_fotografií
```

Pokud je číslo sekvence záporné, nástroj provádí automatickou inkrementaci.

Příklad vstupního souboru .

```
-1; /beach/1362473050.jpg
-1; /beach/1472561701.jpg

-2; /butterfly/114039369.jpg
-2; /butterfly/1357988097.jpg
```

Llf_extract

Nástroj pro extrakci příznaků, umožňuje extrahovat příznaky pomocí SIFT, SURF a FastHOG. Na vstupu uživatel zadá požadovanou metodu extrakce, soubor se vstupními daty a absolutní cestu k adresáři, od kterého jsou relativní cesty ve vstupním souboru předchozího nástroje.

Vocs_create

Tento nástroj vytvoří vizuální slovník tak, že postupně načítá veškeré extrahované příznaky a provádí na nich shlukovou analýzu podle zadaných vstupních parametrů (velikost slovníku, minimální počet prvků ve shluku, počet trénovacích kroků, metoda pro vyhledání nejbližšího středu shluku).

Vocs_dfs

Nástroj, který pro každé vizuální slovo ve slovníku vypočítá inverzní frekvenci dokumentů obsahujících toto slovo. Efektivně jsou tak předpočítány tyto hodnoty, aby byl v dalším kroku umožněn výpočet vizuálního popisku obrázku v jednom průchodu.

Translate

Vytvoří soubor (korpus) obsahující pro každou fotografii vizuální popisek obrázku na základě slovníku, extrahovaných příznaků a parametrů pro vyhledávání a váhování.

Search

Nástroj, který slouží jednak pro vyhledání podobných fotografií vůči dotazované, jednak pro vyhodnocení úspěšnosti vyhledání podobných fotografií stejné třídy.

6.3 Textové popisky

Soubor mapující jména fotografií a k nim odpovídající popisky bylo potřeba nejprve upravit. Program ExifTool extrahuje uživatelem zadané popisky. Pokud tyto popisky nejsou tvořeny pouze malými písmeny a číslicemi bez mezer (tzv. čisté popisky) nebo v případě strojového popisku ve formátu jmennýprostor:predikát=hodnota, následuje za takovým popiskem jeho čistá verze. Proto dříve, než se provede analýza textových popisků, je třeba textový soubor přeformátován tak, aby obsahoval pouze čisté verze popisků.

Takto upravený soubor je ještě transformován pomocí modulu tag_relevance (tento modul provádí výpočet relevance tagů podle algoritmu v kapitole 4.3 s tím rozdílem, že nezahazuje obrázky popsané stejným uživatelem, protože tuto informaci nemá k dispozici) do formátu, kde se první popisek jmenuje sum a udává součet relevancí jednotlivých popisků z důvodu pozdějšího relativního váhování.

Příklad:

```
/photos/1032357425.jpg      sum=1,moms:0,trip:1,england:0
```

Jedním ze vstupních parametrů modulu tag_relevance je počet nejbližších kandidátů, kteří přispívají k relevanci tagu, jak bylo vysvětleno v kapitole 4.3.

6.4 Experimentální aplikace

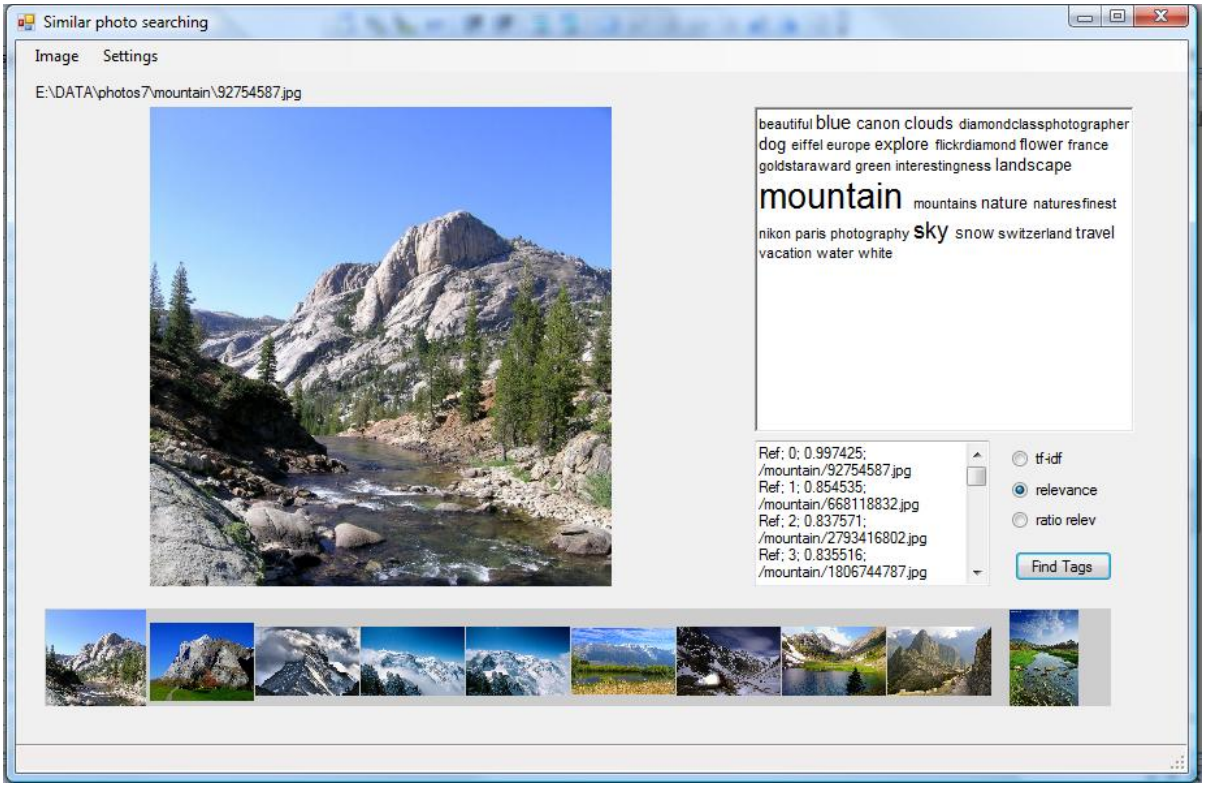
Experimentální aplikace je implementována podle návrhu uvedeného v předchozí kapitole tak, aby splňovala požadavky zadání. Tedy, aby provedla analýzu dat, zobrazila výsledek a umožnila korekci. Aplikace je implementována jako formulářová aplikace (viz Obrázek 13) pomocí vývojového prostředí Visual Studio 2008 firmy Microsoft v jazyce C++/cli a využívá knihovnu pro zpracování obrazu OpenCV a knihovnu VPL.

Pro práci uživatel spustí aplikaci a načte konfigurační soubor, který obsahuje veškeré nastavení potřebné pro vyhledávání podobných fotografií. Jedná se o výběr databáze nebo binárního souboru, ke kterému se uživatel připojí. Dále pak o zvolený korpus dat, nad kterým bude provedeno vyhledávání, textový slovník, stop list, předzpracovaný textový soubor s popisky obrázků včetně relevance jednotlivých popisků, počet nejbližších fotografií, které mají přispět váhou svých popisků k celkovému součtu, limit pro počet navržených štítků a adresář, vůči němuž jsou vztažena relativní umístění fotografií.

Jakmile uživatel načte obrázek, který by chtěl oanotovat, zvolí si kritérium, podle kterého budou počítány váhy textových popisků. Může si vybrat váhování $tf - idf$, které se hodí, když chce štítek, který je hodně rozlišující, tedy nevyskytuje se příliš často, nebo váhování s použitím

natrénované relevance textových popisků. Zde je možnost zvolit absolutní a relativní váhování. Defaultně je vybrána možnost absolutní relevance.

Po kliknutí na tlačítko find tags. Je nad obrázkem provedena extrakce příznaků, jsou nalezena vizuální slova a je vytvořen VpO. Pomocí VpO se vypočte podobnost s obrázky v databázi. Výsledek je seřazený od největší podobnosti. Cyklem se prochází tato posloupnost fotografií a načítají se textové popisky. Ty v případě, že nejsou obsažené ve stop listu, přispívají svojí relevancí (nebo jiným způsobem vypočítanou váhou) násobenou podobností mezi fotografiemi výsledné váze daného popisku. Využívá se datové struktury SortedDictionary, která mapuje řetězec názvů textového popisku na váhu. Pokud je nastaveno, aby se zobrazil jen daný počet výsledných popisků, váhy popisků jsou seřazeny podle velikosti a je nalezena prahovací váha. V poslední fázi se prochází slovník s popisky, porovnává se jejich váha s prahovací váhou. Pokud daný tag má vyšší váhu, je mu vypočítána přímo úměrně velikost písma, kterou bude vytisknut do komponenty richTextBox. Zde pokud se uživateli navržené popisky nelíbí, může je jednoduše vymazat.



Obrázek 13: Experimentální aplikace

7 Experimenty

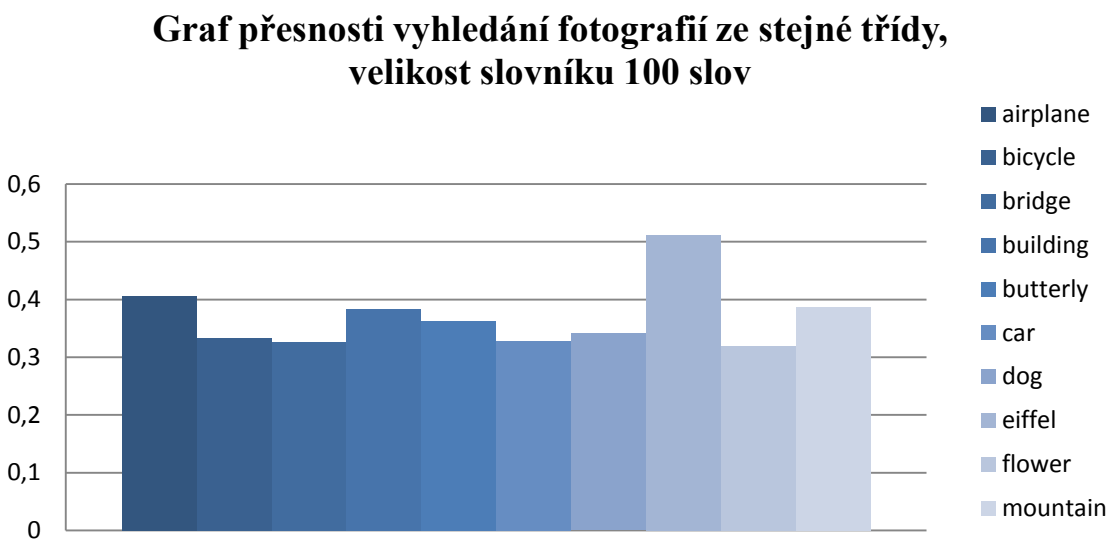
V této kapitole jsou zdokumentovány provedené experimenty. První podkapitola je věnovaná úspěšnosti vyhledání podobných fotografií a ladění parametrů při vytváření slovníku. Druhá podkapitola se zabývá testování přiřazení popisků s různě nastavenými parametry. V poslední podkapitole jsou shrnuty nedostatky aplikace.

7.1 Vyhledávání podobných fotografií

V této podkapitole je zdokumentováno experimentování s velikostí slovníku a jeho vlivu na úspěšnost vyhledávání. Díky tomu, že víme, do které třídy data patří, můžeme provést hledání několika nejbližších výsledků v korpusu a zaznamenávat, zda byla detekce úspěšná. Úspěšnost je poměr mezi počtem nejbližších sousedů ze stejné třídy a počtem vyhledaných sousedů. Testování bylo provedeno postupně na slovnících čítajících 100, 1 000, 10 000 a 100 000 slov. Vždy bylo náhodně vybráno 600 fotografií z databáze a k nim se hledalo 80 nejbližších kandidátu. U každého slovníku jsou uvedeny parametry, se kterými byl natrénován.

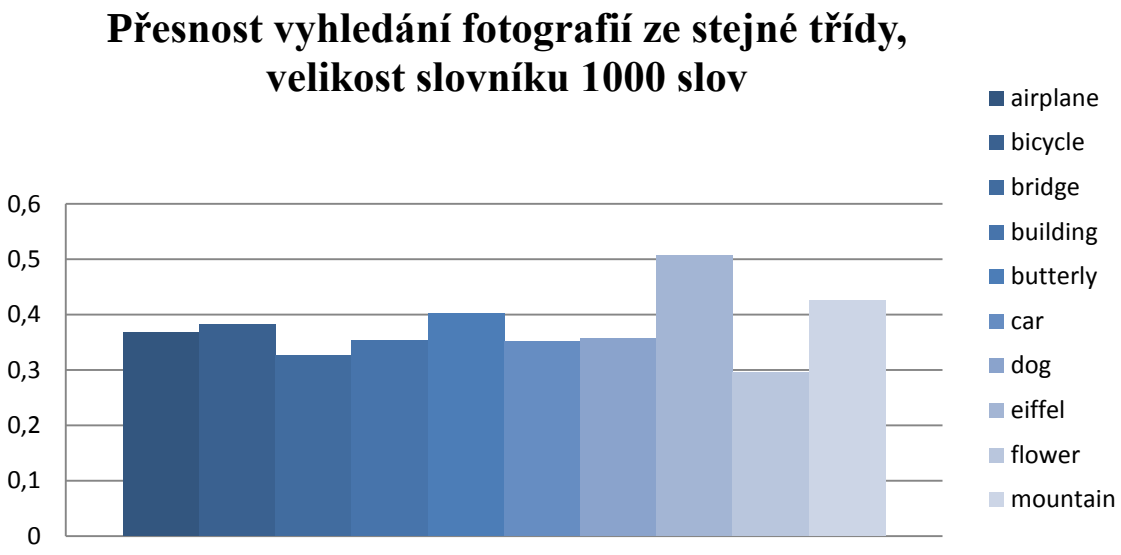
Velikost slovníku – 100

vocs_create.exe -conn bin:data7.tbl -data sift7 -vocabulary "voc_sift_l27 type:kmean clusters:100 min_size:40 search:kd train_steps:10



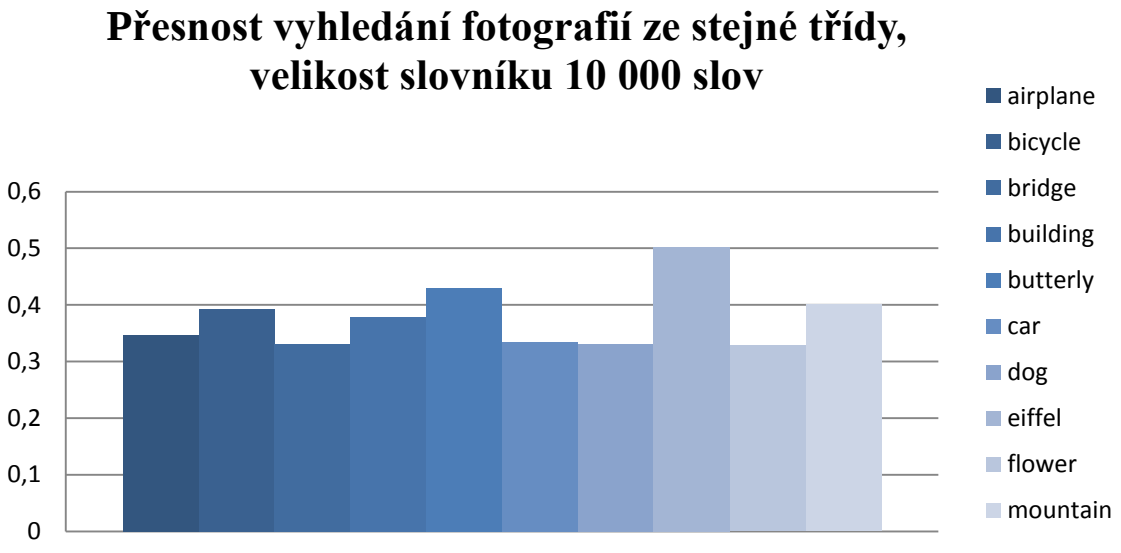
Velikost slovníku – 1 000

vocs_create.exe -conn bin:data7.tbl -data sift7 -vocabulary "voc_sift_l37 type:kmean clusters:1000 min_size:40 search:kd train_steps:10"



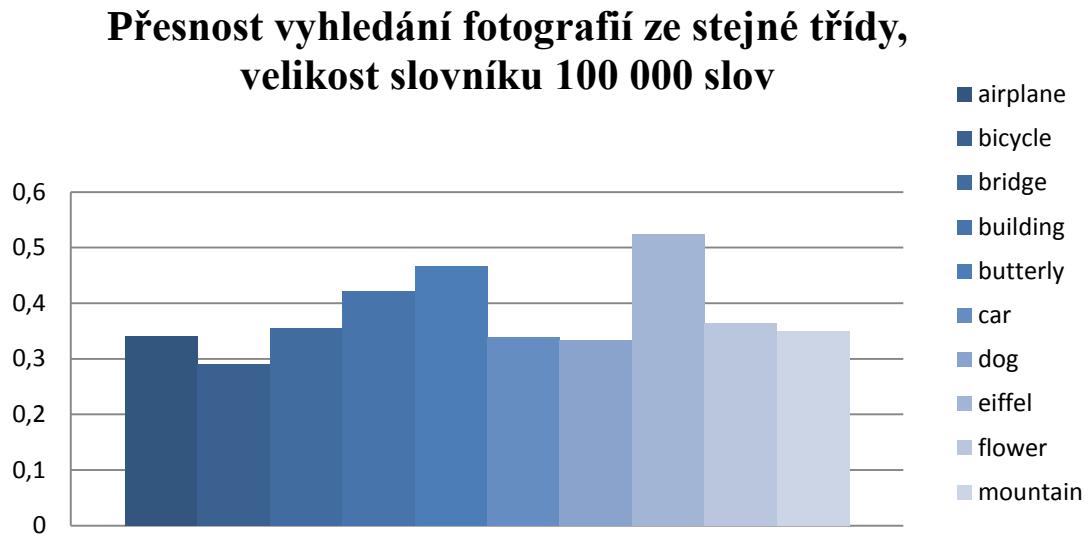
Velikost slovníku – 10 000

vocs_create.exe -conn bin:data7.tbl -data sift7 -vocabulary "voc_sift_l47 type:kmean clusters:10000 min_size:10 search:kd train_steps:10



Velikost slovníku – 100 000

vocs_create.exe -conn bin:data7.tbl -data sift7 -vocabulary "voc_sift_l57 type:kmean clusters:100000 min_size:10 search:kd train_steps:10"



Zhodnocení

Podle předpokladu nejlépe při vyhodnocení dopadla třída obsahující Eiffelovu věž, poněvadž příznaky na ní detekované by měly mít blízké VpO, a proto také malou vzájemnou vzdálenost. Musíme však brát v potaz, že test byl proveden nad relativně malým vzorkem dat. Dobře dopadla také skupina butterfly a mountain. Z výsledků provedeného testování se dá usoudit, že pro oblast našeho využití není nutné trénovat velké slovníky. Zároveň se ukázalo, že vyhledání sousedních fotografií můžeme použít pro výpočet relevance textové informace, čímž dosáhneme kvalitnější nabídky popisků. Kdyby vyhledávání nefungovalo, blížilo by se přesnost vyhledání při dostatečně velkém množství pokusů k hodnotě 0,1 za předpokladu použití pouze testovacích dat, kde je zaručeno, že do některé skupiny patří.

7.2 Testování přiřazení popisků

V této podkapitole je ukázka testování přiřazených popisků. U následujících příkladů byla nejprve vypočítána relevance popisků. Použil se slovník o velikost 1000 slov,

7.2.1 Test 1

Hledalo se sto nejbližších fotografií. Bylo vybráno vždy 5 štítků s největší váhou. Při trénování bylo v úvahu bráno 10, 30 a 50 nejbližších sousedů. Obrázek vlevo tvoří dotaz, obrázky vedle něj odpověď.

Jak moc ovlivňuje při výpočtu relevancí to, jaký zvolíme počet nejbližších sousedů braných v úvahu při trénovací fázi?

relevance

airplane bike car flower mountain

airplane dog flower mountain sky

airplane canon car flower sky

tf-idf

bike bycicle car ferrari systemcollectionsgenericlist1systemstring

ratio

airplane bike bycicle car flower

airplane bycicle car dog flower

airplane bycicle car dog flower

Zhodnocení

Na tomto konkrétním případě, jak poměrová tak absolutní relevance se mění nepatrně. Není tedy třeba nastavovat při trénování velké číslo. Váhování tf-idf se neměnní vůbec protože s touto informací nepracuje. Poslední štítek u tf-idf je kandidát na použití stop listu.

7.2.2 Test 2

Ukázka výsledků, při dotazech obsažených v korpusu. Do výsledku zasahuje 50 nejlepších, velikost slovníku zůstává nezměněna, používá se absolutní relevance.

aircraft airplane blue bulding butterfly bycicle car city flower impressedbeauty insect jet landscape macro monarch nature sky sunset superbmasterpiece yellow

blue clouds diamondclassphotographer eiffel europe explore flower flowers france goldstaraward interestingness landscape mountain nature paris sky snow tower travel vacation

blue canon clouds diamondclassphotographer dog europe explore flickrdiamond flower france interestingness landscape mountain mountains nature sky snow travel water white

aeroplane air aircraft airplane aviation blue california canon car diamondclassphotographer dog flower flying garden jet macro mountain rare sky water

Zhodnocení

Za povšimnutí stojí obrázek s horou, podobné fotografie se našli velmi dobře. Stejně tak navržené popisky.

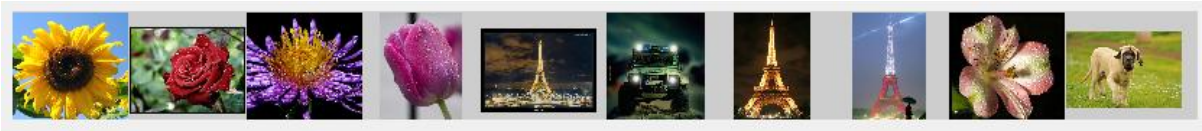
7.2.3 Test 3

Fotografie neobsažené v korpusu. Nastavení zůstává stejné. Předpokládá se, že výsledky budou o poznání horší. Štítky, které bychom si k dané fotografii představovali, vůbec nemusí v systému být anebo jich je velmi málo, takže mají nulovou referenci. Stejně je to s vizuálním obsahem. Pro trénovaní bylo použito velmi malého počtu fotografií.

aircraft airplane anawesomeshot blue bridge bycicle canon car clouds diamondclassphotographer dog flower mountain platinumphoto sky

blue canon car clouds dog explore flower france landscape mountain nature sky snow travel water

airplane blue canon car dog eiffel flower france mountain nature night paris sky tower water



Zhodnocení

Jak bylo uvedeno v úvodu tohoto testu, systém trpí nedostatečným množstvím dat.

8 Závěr

Cílem diplomové práce bylo navázat na semestrální projekt a navrhnout a implementovat experimentální aplikaci, která provede vyhledání vizuálně podobných fotografií zadané fotografii a analyzuje textové popisky u těchto fotografií, dále navrhne ve formě mraku štítků popisky, jež by se měly vztahovat k vizuálnímu obsahu zadané fotografie a umožnit případnou opravu.

Řešení jsem realizoval pomocí knihovny VPL, která poskytuje nástroje pro extrakci příznaků i natrénování vizuálního slovníku. U textových popisků jsem použil metodu pro výpočet jejich relevance, aby byly minimalizovány efekty popisků, které vůbec nesouvisí s obsahem fotografie. Z tohoto důvodu byl také vytvořen stop list. Provedl jsem experimenty zkoumající vliv velikosti slovníku, množství trénovacích dat a různých váhovacích schémat na výsledek.

Nedostatky navrženého systému vidím hlavně v nedostatku trénovacích dat. Zajímavým námětem pro případné rozšíření by bylo vzít v úvahu barevné informace z obrázku a zapojit do hry i globální příznaky, které by společně s příznaky lokálními při shodě dávali větší jistotu správného výsledku. Stejně tak by bylo možné použít metody na redukci sémantické mezery mezi vizuálním obsahem a textovým popisem. Aplikaci by bylo možné dále rozšířit, aby pracovala nad databází.

9 Literatura

[1] MATAS, J., CHUM, O., URBAN, M., PAJDA, T. *Robust Wide Baseline Stereo from Maximaly Stable Extremal Regions* [online]. In proceedings of the British Machine Vision Conference, Cardiff, UK, s 384-393, 2002. [cit. 2009-12-17].
URL <HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/RESEARCH/AFFINE/DET_EVAL_FILES/MATAS_BMVC2002.PDF>

[2] MIKOLAJCZYK, K. et. al. *A Comparison of Affine Region Detectors* [online]. Int. Journal on Computer Vision, 2006. [cit. 2009-12-17].
URL <HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/RESEARCH/AFFINE/DET_EVAL_FILES/VIBES_IJCV2004.PDF>

[3] SONKA, M., HLAVAC, V., BOYEL, R. *Image Processing, Analzsis and Machine Vision*. 3rd edition. Toronto : Thomson, 2008. 829 s. ISBN 0-495-08252-X

[4] Wikipedia. *Harris affine region detector* [online]. [cit. 2009-12-18].
URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/HARRIS_AFFINE_REGION_DETECTOR>

[5] Wikipedia. *Hessian affine region detector* [online]. [cit. 2009-12-18].
URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/HESSIAN_AFFINE_REGION_DETECTOR>

[6] Wikipedia. *Kadir Brady saliency detector* [online]. [cit. 2009-12-18].
URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/KADIR_BRADY_SALIENCY_DETECTOR>

[7] Wikipedia. *Maximally stable extremal regions* [online]. [cit. 2009-12-20].
URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/MAXIMALLY_STABLE_EXTREMAL_REGIONS>

[8] Wikipedia. *Scale-invariant feature transform* [online]. [cit. 2009-12-27].
URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/SCALE-INVARIANT_FEATURE_TRANSFORM>

[9] LOWE, D. *Distinctive Image Features from Scale-Invariant Keypoint* [online]. [cit. 2009-12-27].
URL <HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/RESEARCH/AFFINE/DET_EVAL_FILES/LOWE_IJCV2004.PDF>

[10] FORSSÉN, P., LOWE, G. *Shape Descriptors for Maximaly Stable Extremal Regions* [online]. In Proc. ICCV, 2003. [cit. 2009-12-29].
URL <HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/PUBLICATIONS/PAPERS/SIVIC03.PDF>

[11] Wikipedia. *Image moment* [online]. [cit. 2009-12-29].
URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/IMAGE_MOMENT >

[12] *Affine Covariant Features* [online]. [cit. 2009-12-29].
URL <HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/RESEARCH/AFFINE/DETECTORS.HTML>

[13] Wikipedia. *Singular value decomposition* [online]. [cit. 2009-12-29].
URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/SINGULAR_VALUE_DECOMPOSITION>

[14] SIVIC, J., ZISSERMAN, A. *Video Google: A text retrieval approach to object matching in videos* [online]. In Proc. ICCV, 2003. [cit. 2009-12-27].
URL <HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/PUBLICATIONS/PAPERS/SIVIC03.PDF>

[15] BERAN, V., HEROUT, A., ZEMČÍK, P. *On-line Video Synchronization Based on Visual Vocabularies* [online]. [cit 2010-05-24]. URL <HTTP://WWW.HYPERSCIENCES.ORG/IJSIP/ISS.2-2010/IJSIP-1-2-2010.PDF>

[16] ŠPANĚL, M. *Statistické rozpoznávání a shlukování* [online]. [cit. 2009-12-30]. URL <HTTPS://WWW.FIT.VUTBR.CZ/STUDY/COURSES/POV/PRIVATE/LECTURES/POV_02_STATISTICKE_ROZPOZNAVANI.PDF>

[17] ŽÁRA, J., BENEŠ, B., SOCHOR, J., FELKEL, P. *Moderní počítačová grafika*. Druhé přepracované a rozšířené vydání. Brno : Computer Press, 2004. 609 s. ISBN 80-251-0454-0

[18] PHIBIN, J. et. al. *Object retrieval with large vocabularies and fast spatial matching*. [online]. [cit 2010-05-24]. URL <HTTP://WWW.ROBOTS.OX.AC.UK/~VGG/PUBLICATIONS/PAPERS/PHILBIN07.PDF >

[19] JIANG, Y.,G., NGO, CH., W., YANG, J. *Towards Optimal Bag-of-Features for Object Categorization and Semantic Video Retrieval*. [online].[cit 2010-05-24]. URL <HTTP://WWW.CS.UCLA.EDU/~MJWELCH/MULTIMEDIA/PAPERS/CIVR07_YJIANG.PDF>

[20] LI, X., SNOEK, C., G., M, WORRING, M. *Learning Tag Relevance by Neighbor Voting for Social Image Retrieval*.[online]. [cit. 2010-24-05]. URL <HTTP://STAFF.SCIENCE.UVA.NL/~XIRONG/PUB/MIR08.PDF>

[21]]WANG, X., J., ZHANG, L. *Annotating Images by Mining Image Search Results*. [online]. [cit. 2010-24-05]. URL < HTTP://CITESEERX.IST.PSU.EDU/VIEWDOC/DOWNLOAD?DOI=10.1.1.150.2580&REP=REP1&TYPE=PDF>

[22] Wikipedia: *Tag cloud* [online]. [cit. 2009-12-30]. URL <HTTP://EN.WIKIPEDIA.ORG/WIKI/TAG_CLOUD>

[23] *Flickr* [online]. [cit. 2009-01-04]. URL <HTTP://WWW.FLICKR.COM/>

[24] GERD, J. *Downloader* [počítačový program]. Ver. 2009-11-03. [cit. 2009-01-02]. URL <HTTP://JANTEN.COM/DOWNLOADR/DOWNLOAD.PHP>

[25] SKIJAN, I. *IrfanView* [počítačový program]. Ver. 4.25. [cit. 2010-05-24]. URL <HTTP://WWW.IRFANVIEW.CZ/>

[26] HARVEY, P. *ExifTool* [počítačový program]. Ver. 8.15 [cit. 2010-03-24]. URL <HTTP://WWW.SNO.PHY.QUEENSU.CA/~PHIL/EXIFTOOL/>

[27] BERAN, V. *Video Processing Toolkit*.[cit 2010-05-24]. URL <HTTP://WWW.FIT.VUTBR.CZ/~BERANV/VPF/VPL/>

[28] *OpenCV* [online]. [cit. 2009-01-04].
URL <HTTP://OPENCV.WILLOWGARAGE.COM/WIKI/>

Seznam použitých zkratek a symbolů

MSER	Maximally stable extremal regions
SIFT	Scale-invariant feature transform
VpO	Vizuální popisek obrázku
VPL	Video Processing Toolkit

Seznam příloh

A Přílohy

A.1 Obsah DVD

Stromová struktura adresářů na přiloženém DVD je následující:

- adresář photos – obsahuje trénovací data (fotografie s textovými popisky)
- adresář src – zdrojové kódy programů
- adresář test – zde je připraven dávkový soubor obsahující spouštění jednotlivých modulů a vše potřebné vyzkoušení aplikace
- adresář 3rdparties – programy třetích stran využité při realizaci aplikace
- soubor dp.pdf – tato práce v elektronické podobě
- soubor plakat.pdf – plakát reprezentující tuto práci

A.2 Ukázka dat z třídy eiffel